DMSVIVA 2022

Proceedings of the 28th International DMS Conference on Visualization and Visual Languages

June 29 to 30, 2022
KSIR Virtual Conference Center
Pittsburgh, USA

DOI: 10.18293/DMSVIVA2022

Proceedings preparation, editing and printing are sponsored by KSI Research Inc.

PROCEEDINGS

DMSVIVA2022

The 28th International DMS Conference on Visualization and Visual Languages

Sponsored by

KSI Research Inc. and Knowledge Systems Institute, USA

Technical Program

June 29 to 30, 2022

KSI Research Virtual Conference Center, Pittsburgh, USA

Organized by

KSI Research Inc. and Knowledge Systems Institute, USA

ISBN: 1-891706-55-1
ISSN: 2326-3261 (print)
2326-3318 (online)
DOI: 10.18293/DMSVIVA2022

Additional copies can be ordered from:
KSI Research Inc.
156 Park Square Lane
Pittsburgh, PA 15238 USA
Tel: +1-412-606-5022
Fax: +1-847-679-3166
Email: dms@ksiresearch.org
Web: http://ksiresearch.org/ seke/dmsviva22.html

Proceedings preparation, editing and printing are sponsored by KSI Research Inc. and Knowledge Systems Institute, USA.

Printed by KSI Research Inc. and Knowledge Systems Institute, USA.

DMSVIVA2022

The 28th International DMS Conference on Visualization and Visual Languages

June 29 and 30, 2022

KSIR Virtual Conference Center, Pittsburgh, USA

Conference Organization

DMSVIVA2022 Conference Chair and Co-Chairs

Stefano Cirillo, University of Salerno, Italy; Conference Co-Chair
Yang Zou, Hohai University, China; Conference Co-Chair

DMSVIVA2022 Steering Committee Chair

Shi-Kuo Chang, University of Pittsburgh, USA; Steering Committee Chair

DMSVIVA2022 Steering Committee

Paolo Nesi, University of Florence, Italy; Steering Committee Member
Kia Ng, University of Leeds, UK; Steering Committee Member

DMSVIVA2022 Program Chair and Co-Chair

Bernardo Breve, University of Salerno, Italy; PC Co-chair
Jun Kong, North Dokota State University, USA; PC Co-chair

DMSVIVA2022 Program Committee

Danilo Avola, University of Rome, Italy
Rachel Blagojevic, Massey University, New Zealand
Andrew Blake, University of Brighton, UK
Paolo Bottoni, Universita Sapienza, Italy
Bernardo Breve, University of Salerno, Italy
Loredana Caruccio, University of Salerno, Italy

Maiga Chang, Athabasca University, Canada
WilliamCheng-Chung Chu, Tunghai University, Taiwan
Stefano Cirillo, University of Salerno, Italy
Mauro Coccoli, University of Genova, Italy
Gennaro Costagliola, University of Salerno, Italy
Mattia DeRosa, University of Salerno, Italy
Vincenzo Deufemia, University of Salerno, Italy
Tiansi Dong, Bonn-Aachen International Center for Information Technology, Germany
Martin Erwig, Oregon State University, USA
Larbi Esmahi, Athabasca University, Canada
Edoardo Fadda, Politecnico di Torino, Italy
Andrew Fish, University of Brighton, UK
Daniela Fogli, Universita degli Studi di Brescia, Italy
Manuel Fonseca, University of Lisbon, Portugal
Rita Francese, University of Salerno, Italy
Vittorio Fuccella, University of Salerno, Italy
Ombretta Gaggi, Univ. of Padova, Italy
Pedro Isaias, University of Queensland, Australia
Jonathan Kavalan, University of Florida, USA
Jun Kong, North Dokota State University, USA
Yau-Hwang Kuo, National Cheng Kung University, Taiwan
Robert Laurini, University of Lyon, France
Mark Minas, Universität der Bundeswehr München, Germany
Eloe Nathan, Northwest Missouri State University, USA
Paolo Nesi, University of Florence, Italy
Max North, Southern Polytechnic State University, USA
Michela Paolucci, University of Florence, Italy
Giovanni Pilato, Italian National Research Council, Italy
Giuseppe Polese, University of Salerno, Italy
Michele Risi, University of Salerno, Italy
Peter Rodgers, University of Kent, UK
Domenico Santaniello, University of Salerno, Italy
Michael Wybrow, Monash University, Australia
Weiwei Xing, Beijing Jiao Tung University, China
Atsuo Yoshitaka, JAIST, Japan
Tomas Zeman, Czech Technical University, Czech Republic
Kang Zhang, University of Texas at Dallas, USA
Yang Zou, Hohai University, China

Publicity Chair and Co-Chair

Maiga Chang, Athabasca University, Canada; Publicity Co-Chair
Tiansi Dong, Bonn-Aachen International Center for Information Technology, Germany; Publicity Co-Chair

FOREWORD

On behalf of the Program Committee of the *28th International DMS Conference on Visualization and Visual Languages (DMSVIVA2022)*, we would like to welcome you. This conference aimed at bringing together experts in visualization, visual languages, distance education and distributed multimedia computing, providing a forum for productive discussions about these topics.

It is our pleasure to announce that by the extended deadline of 20 May 1 2022, the conference received 14 submissions. All the papers were rigorously reviewed by at least two members of the international Program Committee, and most of the papers were reviewed by three members of the PC. Based on the review results, 6 papers have been accepted as regular papers with an acceptance rate of 43%. We would like to thank all the authors for their contributions. We also would like to thank all the Program Committee members for their careful and prompt review of submitted papers.

One special feature of this year's conference is that we have arranged to have three highly interesting and relevant keynotes. We thank the three keynote speakers: Professor Gennaro Costagliola, Dr. Gianni.Pantaleo and Dr. Tiansi Dong for their contributions.

We would like to thank the Steering Committee Chair Professor Shi-Kuo Chang for his guidance and leadership throughout organization of this conference. The assistance of the staff at KSI Research and Knowledge Systems Institute is also greatly appreciated, which made the review process smooth and timely.

Finally, we would like to thank you all for joining us in DMSVIVA2022, we really appreciate your participation and your desire to support the community year by year.

Bernardo Breve, University of Salerno, Italy; Program Co-Chair
Jun Kong, North Dakota State University, USA; Program Co-Chair

Table of Content

Session I

Session II

Notes: (S) denotes a short paper.

Keynote

On the Definition and Implementation of Visual Languages

Prof. Gennaro Costagliola
Department of Informatics
University of Salerno
Italy

Abstract

The spread of advanced user interfaces has motivated many researchers in recent decades to focus on studying languages other than traditional string languages. Among them, visual languages are conceived as languages that are received through the sense of sight with a high degree of evocation of the expressed concepts. Consequently, the sentences of visual languages (e.g., Entity-Relation and UML diagrams, Systems Biology Graphical Notation (SBGN)) are usually not restricted to the linear case and are thus expressed in 2D or 3D dimensions without a fixed order in the scanning of their components. These new requirements make the processing of visual languages a complex task that requires new representation models and processing techniques when considered as formal languages.

In this talk I will show how research on visual language representation and parsing has evolved over the years, how the results have been used to support designers in defining and implementing diagrammatic environments and provide future directions for the field.

About the Speaker

Gennaro Costagliola is professor of computer science in the Department of Informatics at the University of Salerno and director of CLUELAB. As part of his research on visual languages, he has published more than 60 papers in journals and conference proceedings covering both theoretical and applicative aspects. These include the definition of new grammatical formalisms for 2D languages, the development of parsing environments for the design and interpretation of diagrammatic sentences, and their application in software engineering. He has served on the editorial board of the Journal of Visual Languages and Computation and is currently co-editor-in-chief of the recently published JVLC. He has also served on the program committee of several conferences on the subject.

Keynote

Enhancing Decision Support Through Big Data Visual Analytics

Gianni Pantaleo, PhD
University of Florence
Italy

Abstract

The increasingly pervasive diffusion of big data (fostered by the advancements in IoT/IoE solutions) and the evolution of visual analytics tools are leading towards innovative data-driven decision support systems. Recent approaches in the implementation of smart solutions for integrating visual analytic tools on top of big data management frameworks and event-driven applications are enabling the development of smarter and more reactive tools, aiming at improving decisional making processes. In this keynote, some insights on the evolution of decision support systems enhanced through big data visual analytics will be presented, especially in the contexts of Smart City and Industry 4.0, as a result of our research in recent years, showing different use cases implemented in several international research projects.

About the Speaker

Gianni Pantaleo is aggregate professor of Computer Science and Information Systems at University of Florence, Department of Information Engineering. His research interests include Artificial Intelligence, Visual Analytics, IoT/IoE and their application in Smart Cities and Industry 4.0. He has been coordinator of several WPs in international R&D projects, and participated in several international research projects, such as Herit-Data Interreg MED, Resolute H2020, Trafair CEF, Sii-Mobility. As a research fellow at DISIT Lab, University of Florence, he collaborated on the development of the Snap4City platform, which was awarded as the best solution in the "Select4Cities" Pre-Commercial Procurement of the European Community to promote open innovation in Smart Cities and IoT/IoE areas.

Keynote

A Geometric Approach to Precise Neural-Symbolic Unification

Dr. Tiansi Dong
University of Bonn
Germany

Abstract

Structure and learning are among the most prominent topics in Artificial Intelligence (AI) today. Structure is traditionally centered in symbolic AI, learning is currently centered in neural networks. How structures and learning can be integrated has been a grand challenge for decades.

In this talk, I show the possibility for a precise unification of symbolic structure and vector embedding. Symbolic reasoning is explainable, rigor, but brittle to noisy inputs. Deep learning overcomes these weaknesses at the cost of the explanability and the rigor of symbolic approach. The unified representation inherits elegant features from both parent sides, namely, explanability, rigor, robust, and trustworthy. This shapes a new way to do AI.

About the Speaker

Tiansi Dong is senior member of ACM, senior researcher and Habilitand at University of Bonn. Since 2015, he has collaborated with the Tsinghua AI group. He jointly published a novel geometric approach to precisely imposing large tree structures onto vector embeddings at AAAI and ICLR in 2019. The method has the potential to bridge the gap between neural and logical reasoning methods, thus having the major impact on the field. He served as the German side PI of Sino-German Symposium “Integrating Symbolic Representation with Numeric Representation for Commonsense Reasoning” in 2019 funded by DFG and NSFC. He was the leading organizer of the Dagstuhl Seminar “Structure and Learning” in 2021 which attracted over thirty interdisciplinary world-leading researchers. His second monograph on Geometric Approach to Precise Neural-Symbolic Unification was published by Springer in 2021.

Digital Twin Framework for Smart City Solutions

L. Adreani[1], P. Bellini[1], C. Colombo[2], M. Fanfani[1,2], P. Nesi[1], G. Pantaleo[1], R. Pisanu[2]
University of Florence, Florence, Italy, email: <name>.<surname>@unifi.it
1) DISIT lab, https://www.disit.org, https://www.snap4city.org
2) Computational Vision Group http://cvg.dsi.unifi.it/cvg/

***Abstract*—Recently, 3D city modelling has attracted a growing interest as a building block for creating city Digital Twins. They are complex representations that include interactive representations of buildings and infrastructures, integrated with the wide range of data typically useful in a Smart City environment. This paper presents an automatic method for producing 3D city models from a various set of data, as well as its integration into the open-source Smart City framework, Snap4City. The proposed solution offers a method for creating effective integrated data visualizations of 3D city entities coupled with a large variety of Smart City data (e.g., IoT Devices which generate time-series data, heatmaps, geometries and shapes related to traffic flows, bus routes, cycling paths). The solution is based on a deep learning approach for rooftop detection and alignment based on a U-Net architecture. The implementation has been enforced into the open-source Snap4City Smart City platform, and has been validated by using a manually created ground-truth of 200 buildings scattered uniformly in the central area of Florence, plus a number of meshes representing a number of facades (not detailed in this paper), and traffic flows, pins, heatmaps, etc.**

Keywords—3D City model, Photorealistic texture, digital twin, Smart City applications.

I. Introduction

Smart Cities are complex infrastructures integrating multiple linked data sources, Internet of Things (IoT) devices and applications, involving many different data and stakeholders. In this context, spatial data information may act as enabler for smart applications and decision support systems, provided that they are interoperable with legacy and future solutions [1]. Recently, the aspects related to digital 3D city modelling and digital twin have gained a growing interest, since they allow to create a more realistic context in which the decision makers can perform analyses, simulations, planning and monitoring in several different domains and application areas (e.g., urban planning, energy management, traffic and mobility, disaster management, air pollution monitoring). Many approaches have been proposed in literature, such as: CityGML, CityJSON, the combination of Building Information Modeling (BIM) and Geographic Information System (GIS) providing a City Information Modeling (CIM) [2].

In the past years, a relevant amount of research has been made in the field of 3D city modelling, to recreate realistic visualizations. However, due to the typical size of a city, handling all the data and their processing is a challenging task still remained unsolved [4]. One critical aspect of developing a high-fidelity 3D city model is to find the correct model and format for the data that can be rendered by a visual interface and may be on web browser. For this purpose, a set of requirements have been proposed by CityGML, according to different levels of detail (LoD) which can be addressed by the models. According to [3], there are five levels of detail: LoD0 is represented by those models having only a 2D map with 3D terrain; LoD1 presents buildings as simple boxes; LoD2 adds rooftops details to LoD1 buildings; LoD3 presents also external facades structure; LoD4 adds building interiors. LoD4 was introduced in CityGML 2.0, but it was removed in the latest version of CityGML 3.0. The CityGML and CityJSON have defined a format for the representation of geometry and topology for 3D buildings, using respectively XML and JSON. CityGML 3.0 integrates a BIM standard, alongside the GIS (Geographical Information Systems) format, from Industrial Foundation Class (IFC) [5]. Some integrations of CityGML have been proposed in real cases, such as the city of Helsinki, in which a LoD3 city model was implemented and made publicly available [6]. However, the system do not provide integration with IoT data or other kind of city data. Another similar integration was made by the city of Rotterdam [27], recreating a LoD2 type of buildings, however neither integrating any decoration elements nor elevation of terrain (this is relevant aspect for non flat cities). An attempt of making a LoD3 3D city model was made by ETH Zurich with the VarCity [7]. However, the provided semantic information is generally limited to a small number of semantic classes. The **3dcitydb** implements a 3D model for the city of Berlin [28], providing a pickable model of LoD2 buildings, supporting also WMS (Web Map Service) layers (typical of GIS solutions providing maps, heatmaps and orthomaps) and terrain layer. However, neither of those are provided. The city of Stockholm [29] implements many aspects of Digital Twin concept, such as POI (point of interest), LoD3 type buildings, either with 3D tiles and a modelled one, and others 3D entities. However, the solution lacks in the implementation of any WMS heatmap.

In the context of 3D city data collection, advancements in Light Detection And Ranging (LiDAR) technology allow to model urban topography at spatial resolution and granularity which were not achievable before the advent of this technology [8]. In [9], a method to create a city model from a point cloud generated by LiDAR technology is presented. This approach has shown to reduce the time to generate the model, but it cannot process unsymmetrical objects and presents some geometrical error.

A more pleasant and realistic 3D city representations can be obtained by enhancing them with textures extracted from RGB images. In particular, rooftops textures can be obtained from orthomaps or satellite images, facades image patterns, etc. However, this is not an easy task: at first, rooftops have to be detected in the RGB images [10]; then, the segmented patches must be carefully aligned with the top-view of the 3D map.

DOI reference number: 10.18293/DMSVIVA22-012

Indeed, even if geolocalization information is typically available, errors are present due to uncertainties [11] and an accurate multi-modal registration is required (e.g., between the RGB images and the 3D structure) [12]. In the literature, several works have addressed these topics using both computer vision standard and learning-based solutions. In [13], handcrafted features and a hierarchical segmentation approach have been used to identify the buildings in rural areas. SVMs (support Vector Machines) [14] and Random Forests [15] have also been used to address this task. For example, in [16] the authors proposed a three-steps method based on color-based clustering, roof detection using an SVM and a final false negative recovery. Slightly different, in [17] a pair-wise exploitation of satellite images has been used to reconstruct a 3D model that could then be employed to identify rooftop regions. However, such solutions not only have some limitations when working on areas with dense buildings, but also require a successive registration on the 3D map. More recently, deep learning based solutions appeared for remote sensed image processing [18], [19]. In [20], a Mask R-CNN (region based convolutional neural network) [21] was used to detect rooftops from aerial images. Differently, in [22], [23] a U-Net architecture [24] has been preferred. Moreover, these last two solutions provide not only rooftop segmentation but also the registration on 3D data.

In this paper, a 3D City Modelling Framework for Smart City Digital Twin with textures is presented. The main contributions of this paper are the following: first, the production of a full functional solution showing 3D city representations on the basis of roads, building planimetry, high, detailed buildings with meshes. Thus the creation of a full automatized algorithm to map the buildings in the area, creating building models from building type and the integration of terrain aspects/pattern, more sophisticated 3D shapes based on meshes. Thus the integration of the 3D city representations into a Smart City framework (the open-source Snap4City platform), in order to provide a smart environment and applications for visualizing city entities and related data (coming, for instance, from IoT devices generating time-series data, heatmaps, geometries and shapes related to traffic flows, bus routes, cycling paths etc.), with the possibility to pick single city elements or buildings on 3D city representation, and inspect their data and attributes. The proposed solution is an open-source web-based tool for producing a global digital twin integrating IoT and many other kind of Smart City data, which has been designed to satisfy the most of the identified requirements as reported in the paper.

The paper is organized as follows: in Section II, requirements are presented. In section III, the architecture is described putting in evidence the data flow. The model is detailed in Section IV. In section V, details of the image processing solution based on machine learning is presented for producing the roofs' patterns from orthomaps. In Section I, the final result and process is presented. Some notes above the distribution of 3D information presented is described in Section VI. In Section VII, some notes on the validation regarding the process for roof patter estimation are reported. Finally, conclusion are drawn in section VIII.

II. Requirements Analysis

With the aim of creating a Digital Twin in the context of smart cities, the 3D representation of buildings in the city covers a relevant role. To this end, a set of specific requirements have been identified and are reported in this section. In the past a similar approach has been proposed by CityGML which defined different levels of detail (LoD) for the models [3]. The CityGML approach was mainly on the visual represented and it is actually not enough detailed to describe the needs of full Digital Twin models in the Smart City solutions for decision makers. *Therefore, a more complete set of requirements and an assessment model for Web delivering of 3D representations of Digital Twins at the support of a decision support system is presented in this section.* Most of the requirements are related to the 3D representation and to the integration of 3D data with the massive data infrastructure in back which actually supports the decision makers. For example, to move a bus stop, to close an area for a market, to see the impact of some event. In particular, the solution has to provide support for representing in the 3D context, the:

R1. **buildings of the city as city structure**, roads, gardens, etc. The single building should be represented with realistic details in terms of shape (facades, roof, towers, cupolas, etc.), and patterns on facades and roofs. To this end, different techniques can be adopted to model the buildings. For example, (i) the simple bounding box of the buildings obtained from the perimeter extruded up to the heights of the eaves, (ii) the creation of meshes precisely describing every tiny detail of the physical structure.

R2. **ground information** as road shapes and names, names of squares and localities, etc., exploiting the so called Orthomaps, with eventual real aerial view patterns. They are typically provided in terms of multi resolution tiled images from GIS systems using WMS protocol;

R3. **one or more heatmaps** superimposed (and transparent) on the ground level information without overlapping the buildings. For example, to represent some information, such as: the heatmaps of temperature, traffic flow, pollutant, people flow, etc. Also in this case, they are typically provided in term of multi resolution geolocated tiled images, provided by GIS using WMS protocol;

R4. **paths and areas** super-imposed on the ground and on heatmaps levels without overlapping the buildings, for example those needed to describe the perimeters of gardens, the cycling paths, the trajectories, border of gov areas, etc. This information is quite specific and has to be produced on the basis of the information recovered from some Open Data. Once recovered it can be distributed by using GIS in WFS/WMS protocols;

R5. **pin marking the position** of services, IoT Devices, Point of Interest, POI, Key Performance Indicator, KPI, etc., and providing clickable information according to some data model which may provide access to Time Series, shapes, etc. This information is quite specific and can be produced on the basis of the information recovered from Private and/or Open Data;

R6. **terrain information and elevation**, so that the skyline of the city may include the shape of eventual mountains around, and under the city as well. This also means that the buildings and Orthomaps should be placed according to the terrain elevation.

R7. **additional 3D entities** for completing the realism of the scenario, such as: trees, benches, fountains, semaphores, digital signages, and any other city furniture, etc.

	CityGML [3]	**Helsinki [6]**	**Rotterdam [27]**	**Berlin [28]**	**Stockholm [29]**
R1.i	Yes (LoD1)	No (only available in higher detail)	No (only available in higher detail)	No (only available in higher detail)	No (only available in higher detail)
R1.ii	Yes (LoD3)	Yes (either with object or 3D tiles)	Yes (LoD2)	Yes (LoD2)	Yes (LoD3)
R2	No	Yes (C)	Yes (C)	Yes (C)	Yes (but with a fixed Orthomap)
R3	No	No	No	Yes (does not include Wms)	No
R4	No	Yes (C)	Yes (C)	No (x)	Yes
R5	No	No	No	No	Yes
R6	Yes	Yes (with 3D tiles)	No	No	Yes
R7	Yes (LoD2)	Yes (with 3D tiles)	No	No	Yes (3d tiles and single entity)
RA	No(*)	Yes	Yes	Yes	Yes
RB	No(*)	No	No	No	No
RC	No(*)	No	No	No	Yes
RD.1	not clear (may be)	Yes (when models are loaded as object, not if loaded as 3D tiles)	Yes	Yes	No
RD.2	No	No	No	No	No
RE	No(*)	No	No	No	Yes
RF	No(*)	Yes (**)	Yes (**)	No (x)	No

Table 1 -- Comparison of 3D representation platforms for Digital Twins vs Smart City. Where: (*) defines only the building model, (**) functionality implemented in CESIUM but without any model placed underground, (x) use CESIUM, it could be possible to integrate, (C) based on CESIUM.

In addition, the solution has to be capable to provide some interactivity on the above mentioned 3D data structures, in particular it should be capable to depict the 3D scene:

RA. according to the point of view, providing capabilities for changing it by: zoom, rotate, tilt, and pan the scene and also changing the light or time of the day/night (this may lead to produce shades), etc.

RB. with the sky, maybe with different sky conditions according to the actual day, light condition, weather, or weather forecast.

RC. providing access to the information associated with augmenting PINs: POI, KPI, etc., and maybe to real time data, and time series associated with eventual IoT Devices located on the 3D scene.

RD. providing the possibility of selecting each single building to: (1) pass at a more detailed information associated with the building, or (2) go into a BIM view of the building, with the possibility of navigating into the building structure, and again to access at the internal data associate to PINs into the building. May be also disabling the building view to see only the 3D of city without the buildings but with PINs.

RE. providing possibility of selecting an element (3D, PIN, ground, heatmap) to provoke a call back into a business logic tool for provoking events and actions in the systems, at which the developers may associate intelligence activities, analytics, other views, etc..

RF. providing the possibility of inspecting the ground terrain and see the detailed 3D elements placed in the underground, such as water pipes, or located in the ground as benches, luminaries, red lights, etc.

According to the identified requirements, in the following **Table 1,** an assessment of the most relevant solutions is reported.

III. Architecture and Process

According to the above described requirements, a solution for Smart City Digital Twin, **SCDT**, has to address three main aspects: **(a)** the **3D model** enabling the representation of the information in integrated manner, **(b)** the **software architecture for distributing** and provide access to the 3D representation via a suitable user interface presenting the (a) 3D model including the interactivities features, and **(c)** the **production process** of the 3D models by starting from multiple information which have to be recovered from accessible resources or produced/acquired,

The above described requirements from R1 to R7 mainly impact on (a) and (b) for the resulting performance on distribute and reproduce the representation in real time on browser. Thus, providing support for the users to interact with the 3D representation in real time. The system presents challenging aspects due to the large amount of data to be processed on client side on the basis of the point of view. This impacts especially when several details are provided at the same time in the same view, e.g., photorealistic textures, detailed heatmaps, complex terrains shape that implies to compute several projections to avoid overlaps, etc. In these cases, the issue is typically mitigated at the expense of a lower resolution of textures.

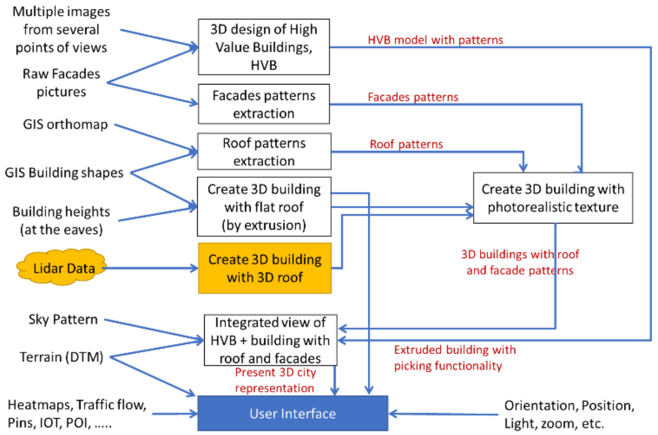

Multiple images from several points of views
Raw Facades pictures
GIS orthomap
GIS Building shapes
Building heights (at the eaves)
Lidar Data
Sky Pattern
Terrain (DTM)
Heatmaps, Traffic flow, Pins, IOT, POI, ...
3D design of High Value Buildings, HVB
Facades patterns extraction
Roof patterns extraction
Create 3D building with flat roof (by extrusion)
Create 3D building with 3D roof
Integrated view of HVB + building with roof and facades
User Interface
HVB model with patterns
Facades patterns
Roof patterns
Create 3D building with photorealistic texture
3D buildings with roof and facade patterns
Extruded building with picking functionality
Present 3D city representation
Orientation, Position, Light, zoom, etc.

Figure 1 – Data Flow of the production process for creating a Digital Twin for smart cities.

On the other hand, the features from RA to RF have to be mainly satisfied by the production process © of the data model to be distributed according to (a) and (b). In fact, the model can be composed by several elements: 3D representation, meshes, patterns, etc. The process to pass from images and data to the integrated 3D model is not trivial as partially described in this paper for some aspects. On this regard, the production process to produce the **SCDT** model is depicted in **Figure** 1. The production process puts in evidence the data sources: GIS, raw images, building shapes, heatmaps, PINs, POI, IoT devices, Terrain DTM (Digital Terrain Model), etc., and the optional LIDAR data which may be exploited for adding details and shortcutting some of the procedures.

According to **Figure 1**, the production process for the creation of the 3D model requires a set of sub-processes:

- **Roof pattern extraction**: photorealistic textures of building rooftops can be obtained from orthomaps. Since orthomaps are typically roughly geo-localized, a careful registration w.r.t. the building shapes is required. After that, textures can be extracted and provided as PNG or JPEG files.
- **Facades pattern extraction:** differently from rooftop textures, where the used orthomaps are relatively easily accessible nowadays, façade texturing requires a specific acquisition campaign. Moreover, the acquired RGB images must be processed to remove radial and projective distortions, and finally, the building facades must be accurately identified and extracted. As for rooftops, obtained textures can be provide as PNG or JPEG files.
- **Create 3D buildings with flat roof (by extrusion):** given the building shapes plus their height, typically measured at their eaves, simple 3D models with flat rooftop can be obtained. The resulting data format is a GeoJSON file with a height/elevation attribute to compute the building extrusion from the ground at run time. This is the model used to implement the picking functionality.
- **Create 3D building with 3D roof**: when a Digital Surface Model (DSM) is available, obtained from LIDAR data or other acquisition modality, accurate 3D roof shapes can be obtained to build a more realistic SCDT. The buildings 3D models can be provided as glTF (GL transmission format) files, with geo-localization information.
- **Create 3D building with photorealistic textures:** the 3D buildings obtained by extrusion or exploiting a DSM can be enhanced with photorealistic rooftop and façade patterns by applying textures extracted from RGB images. Textured building models are saved in glTF files, with geo-localization information.
- **3D design of High Value Buildings, HVBs**: in order to produce accurate representation of HVBs a manual 3D design or automatic computer vision techniques (such as Structure from Motion) can be employed. This requires precise measurements or specific image/video acquisition campaign. Additionally, geo-localization information must be provided. Also in this case, the resulting textured 3D models can be exported as geo-localized glTF files.
- **Integrated view of HVBs + buildings with roof and facades**: the building 3D models and the HVB models are finally placed into a unique 3D representation exploiting their geo-localization information, thus obtaining the complete 3D representation for the SCDT.

The general architecture for distributing **SCDT** includes a set of data integrating 3D models, meshes, with DTM, heatmap, traffic flow, Pins, IOT, POI, etc., as described in the paper.

For the distribution of the data:

- **3D representation File in GeoJSON via HTTPS**: it describes the 3D structure of the city and all information related to it. It is used to represent the city model in extruded mode and to retrieve the buildings information or other BIM data for the picking functionality.
- **3D representation File in glTF/GLB (GLB is the binary version of glTF) via HTTPS**: it describes the 3D structure of the city in terms of building and their relationships with the other graphic elements: facades, meshes of HVB, textures and materials.
- **Pattern files via HTTPS**: pattern images for facades, roofs, DTM files in PNG format, Sky texture, etc.,
- **GIS server via WMS** over https is providing (via GeoServer, also integrated into Snap4City platform): orthomaps, maps, heatmaps, animated heatmaps, traffic flows, animated traffic flows, etc., on the basis of the portion of the map shown in the window frame.
- **SuperService Map of Snap4City platform via smart city API** via https [33], [30] is providing semantic details in JSON such as: roads graph, POI, IoT data, Pins, cycling paths, vectorial traffic flows, etc., on the basis of the portion of the map shown in the window.

IV. Model and Representation

The model for creating the 3D representations, which allows to provide all the above mentioned information, is based on a hierarchical layered structure depicted in **Figure 2** and described in this sections.

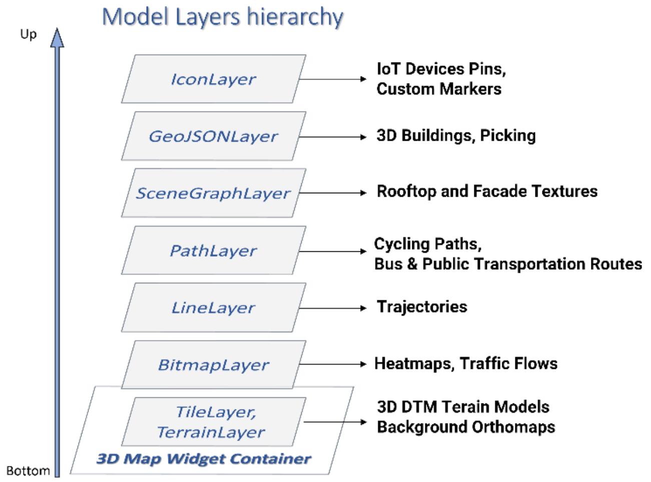


Figure 2: Hierarchical layers structure of the model

The layered solution has been implemented via WebGL API, in order to process all the data in parallel, thanks to the GPU passthrough, to this end, the open-source library called Deck.gl has been used. All the layers needed for the representation of the

Snap4City platform data types have been implemented, and they are loaded at runtime on user demand. Thanks to the multi-layer structure of deck.gl, layers have been implemented individually with their own safe context, to avoid interferences one with each other. Every layer has its own scope, managing its own data type. Therefore, in the following we are introducing the implemented layers to describe data types provided in the Snap4City 3D representation.

First, the base deck application has been realized by using a custom implementation and management of the viewState object, in which all the geographical information for the map (such as latitude, longitude, zoom, etc.), are defined. We have also implemented a custom rendering in order to add features like SkyBox that need direct access to the WebGL context. Starting from the first layer. The elevation of the terrain has been modelled by implementing a composite layer called TileLayer, which is used to divide the maps in multiple tiles with their own sublayer: for each tile a sublayer called TerrainLayer has been created. Thus the elevation map, in the form of DTM files, has been used to create the TerrainLayer 3D model from the map, and the background orthomap has been used as a texture of the terrain objects. The result is a 3D representation of terrain with texture to better represent the territory.

The background orthomaps have been also implemented through a TileLayer. In this case, we used the BitmapLayer to display an image in the map. This method has been also used to represent heatmaps, which are essentials to provide a fast access / representation to large amounts of data. In order to implement heatmap visualization in deck.gl, we used the composite layer which automatically retrieves heatmaps from a dedicated geo-server (through several formats, including WMS) and displays them as an image. Heatmaps can be static or animated; static heatmaps are viewed as single PNG images, while animated ones are sent by the geo-server in GIF format, and they are later divided in multiple images and rendered sequentially with a customizable delay time.

For the implementation of data coming from different sources like IoT devices, trajectories, cycling paths, etc., various layers with a specific JSON mapping have been implemented. To display paths and geometries, different layers depending on the type of geometry to be displayed have been used, i.e. LineLayer for trajectories, PathLayer for the cycling path. IoT devices are also displayed as pickable markers on the map. When a user selects one of them, a popup with the sensor information (static attributes as well as real-time data, if available) is shown. Whenever the sensor provides real-time data, they can be displayed on dedicated widgets, such as time trends, when the user requests them.

3D representation of buildings are provided in two manners: Extruded and Realistic (meshes, HVB). Extruded buildings are implemented by using a GeoJSON file, in order to have a faster loading time, and this is required because this type of buildings are loaded even when the realistic ones are loaded.

Realistic buildings HVB (presenting photorealistic rooftop details and eventually facades textures) can be loaded as both SceneGraph and 3D tiles. In order to implement the picking functionality we need also to render the extruded buildings underneath. The Extruded type is totally described in a single GeoJSON file, where the following elements are defined for each building: the base polygon, the height, and various other attributes and information. The GeoJSON file is loaded in a layer called GeoJSONLayer, and it is responsible to take all the features in the file and display them on the map, with the base polygon extruded by its height. In the case of Realistic building data type, we use the glTF and GLB formats to describe the scene, and they are loaded by the SceneGraphLayer. This type of integration works well to achieve impressive visualization without impacting too much on the application performances. 3D buildings can also be individually picked on map, in order to see all the building information, besides linking to dedicated BIM representations or other details, if available.

V. Production Process

In this section, we present details of our implemented subprocesses to (A) extract roof patterns, (B) create 3D building with flat roof and photorealistic textures, and (C) integrate HVB and 3D building into a unique 3D representation.

A. *Roof pattern extraction*

Orthomaps of the city of Florence, kindly provided by the “Sistema Informativo Territoriale ed Ambientale” of Tuscany Region was used to obtain the roof’s textures. These RGB photos are tiles with a resolution of 8200x6200 pixels, with partial overlap and rough geo-localization in the EPSG 3003 (Monte Mario / Italy zone 1) coordinate system.

To start with, the aerial images and the 2D GIS building shapes (expressed in the EPSG 4326 coordinate system (Geodetic Parameter Dataset, Originally created by European Petroleum Survey Group)) were converted into a common coordinate reference system. We noticed that by merely translating the orthographic photos from EPSG 3003 to EPSG 4326 was not convenient, as it produced evident alterations in the Ground Sample Distance (GSD, i.e., is the distance, in meters, between pixel centres measured on the ground). To mitigate this effect and better maintain the GSD, we selected a third common coordinate system (EPSG 3857 – WGS84/Pseudo-Mercator) onto which to project both images and shapes.

Multiple orthomap tiles describing the considered area were fused into a single mosaic image using the Geospatial Data Abstraction Library, GDAL (https://gdal.org/). Then, we down sampled the mosaic image by a factor of 1/4. This size reduction was crucial in order to obtain a relevant speed-up in the successive steps, yet without losing accuracy, as the chosen image resolution allows the rooftop detection and alignment deep net (see hereafter) to operate optimally.

To detect the rooftops from the orthomaps and align them with the building shapes, we used the method presented in [23], based on a double U-Net architecture exploiting multi-resolution [25] and multi-task learning [26]. The net takes as input an RGB orthomap and the corresponding cadastral map (represented as a binary image), and outputs a list of multi-polygons aligned with the RGB image. In order to obtain the cadastral map, the 2D shapes of the buildings were converted into a raster binary image. The output multi-polygons, up-scaled to take into account the image down-sampling previously done, were then exploited to both extract rooftop textures (from the full resolution mosaic) and align them with the 2D building shapes. An affine transformation to warp the mosaic Orthomaps and register it w.r.t. the 2D building shapes was computed. However, using a single transformation for all the multi-polygons would give rise to local inaccuracies. For this reason, we computed a dedicated transformation for each multi-polygon and locally warped the image so as to obtain a better registration. Specifically, given the vertexes of an aligned multi-polygon V_A and the vertexes of the corresponding 2D shape V_S an affine transformation T was estimated such as

$$V_S = TV_A \tag{1}$$

Then, according to the estimated T, the orthomap was warped and the considered rooftop was extracted. After repeating this process for all the multi-polygons, a complete warped orthomap (including only the rooftops) was obtained and exported as JPEG file. Note that, while exporting the texture image, different resolution can be used to obtain smaller weights and faster visualization.

B. *Creation of 3D model with flat roof and photorealistic textures*

3D model construction and texturing were carried out with *Blender*. The building 3D models were obtained by extrusion from the 2D shapes exploiting their height attributes (included in a GeoJSON as above described) with the *BlenderGIS* library. Then a UV-map of the roof areas was created by retrieving the surfaces with normal vectors perpendicular to the main plane, and the warped orthomap was used to texture the polygons described in the UV-map using the *Python Blender API*.

C. *HVB integration*

Using *Blender*, we were also able to include and geo-locating in the map the 3D models of HVBs. For example, as shown in **Figure** 3, an accurate 3D reconstructions of Santa Maria del Fiore Cathedral (Florence Dome) was placed into the 3D representation, thus achieving a nicer final result.

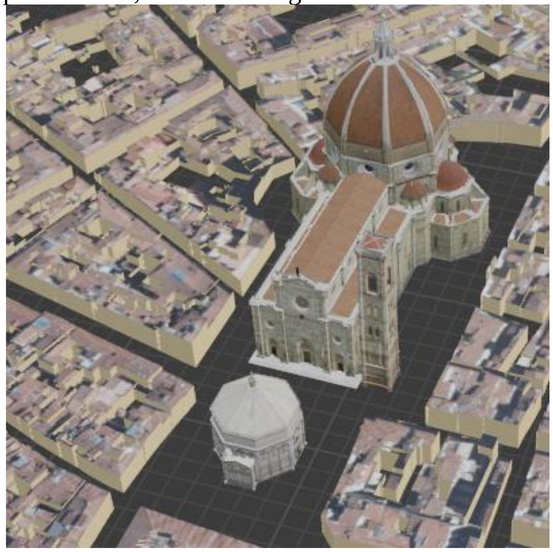


Figure 3: An example of integration of a HVB into the 3D map (in this case the Santa Maria del Fiore Cathedral in Florence).

The obtained 3D textured models of the buildings as well as the HVB models were exported in **glTF** format (including 3D geometries, textures, and coordinates) ready to be deployed in the Snap4City platform using the *SceneGraphLayer* of the *deck.gl* framework (https://deck.gl/).

VI. Access and Distribution in Snap4City

Snap4City is an open-source platform developed at DISIT Lab, University of Florence (https://www.snap4city.org/), [30] [31], [32]. The platform manages heterogeneous data sources, such as: IoT devices (city sensors and actuators, as well as private devices, supporting a large variety of brokers and protocols), open data, external services. For each different kind of data, static attributes (such as geographical information and other metadata) and also real-time data (when available) are collected. Device data are semantically indexed in an RDF Knowledge Base, thus they can be retrieved by dedicated APIs and exploited by Data Analytics processes and IoT applications to perform analyses, simulations, forecasts etc. This allows users to produce new knowledge on data, which can be shown on user interface through Dashboards and a wide range of widgets (showing data both in pull and push modalities). The purpose of integrating the photorealistic 3D city model obtained with the method described in Section IV into the Snap4City platform is to provide a Multi-Data map which can allow the visualization of an interactive 3D environment of the city, with the possibility of inspecting the different kinds of entities and related data, such as: IoT devices, Points of Interests (POI), heatmaps, geometries related to bus routes, cycle paths, traffic flows, etc. In this way, the Snap4City platform allows to exploit a complete open-source framework that can collect, process, and manage all the data needed to obtain a high-fidelity Smart City Digital Twin.

In order to integrate the 3D representations in the Snap4City platform, the deck.gl open-source library has been used, as described in Section V. By exploiting the multi-layer structure of deck.gl, we implemented a distinct layer for every type of data supported by the platform. All layers can be viewed and removed dynamically by user choice. An example of the resulting 3D map is shown in **Figure** 4: the 3D representations can be instantiated by users as a customizable widget in their own dashboards. **Figure** 4 represents the 3D city representation with the addition of textures and 3D model enriched with the textures obtained using the method described in Section V, the model presented in Section IV and the whole architecture of Section III. The tool is freely accessible on web and also includes heatmaps, traffic flow sensors, traffic flow data, animations, PINs for IOT and POI, etc.

Regarding the implementation in deck.gl, first an IconLayer was implemented to represent all the IoT devices managed by the platform. IoT devices are ingested and stored in a semantic Knowledge Base, and they are classified by semantic categories. Therefore, a pool with different icons for each type of device category is used to represent device markers on map. The user can access to all information given by a specific sensor and city element by simply clicking on the device PIN; in this way, a popup is shown presenting static attributes and, when available, real-time and historical data can be selected and viewed on dedicated time-trend and single-content widgets.

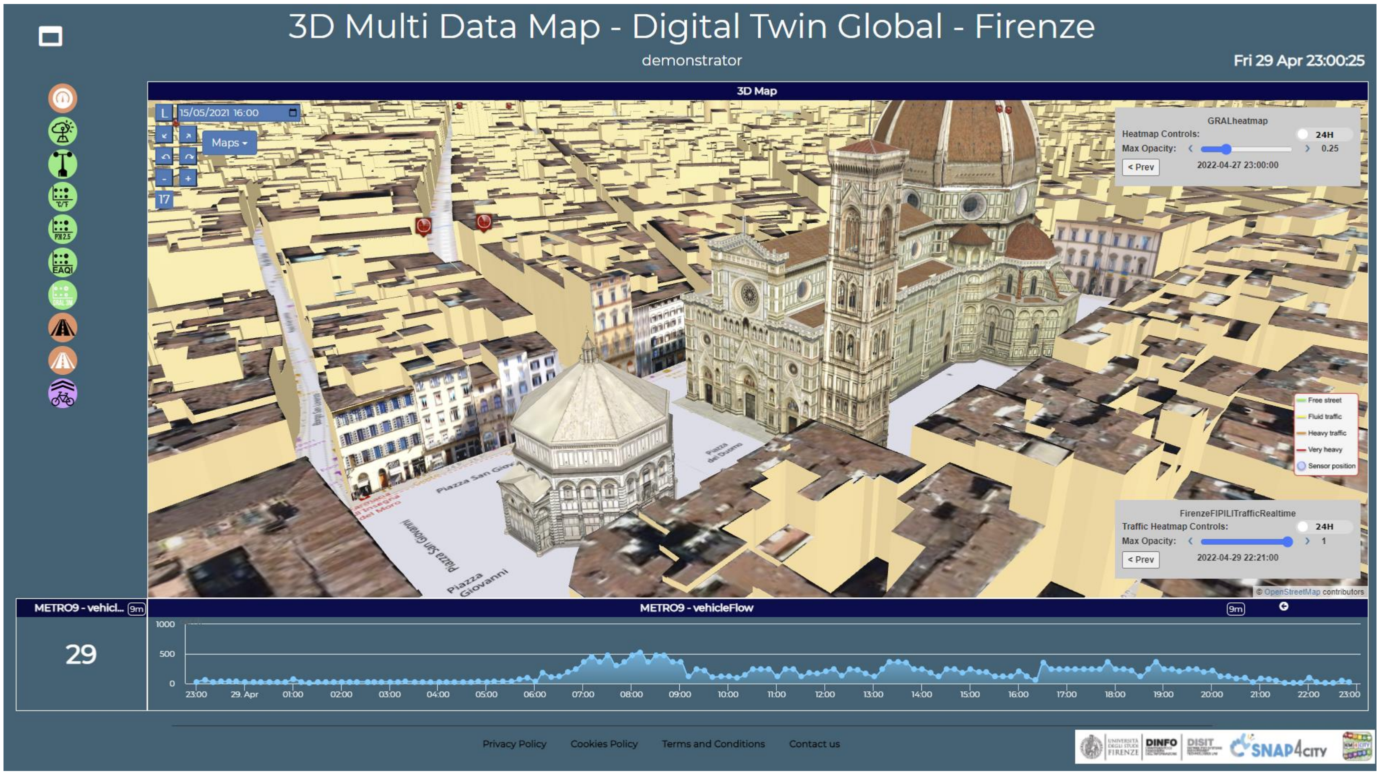


Figure 4: 3D Multi Data Map of Snap4City with addition of textures and mesh based 3D building (the Florence dome) [31], [32]. https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MzI5Mw== accessible to all.

The code of the open source Snap4City Dashboard Builder is available at the following GitHube repository: https://github.com/disit/dashboard-builder.

VII. Rooftop extraction validation

To obtain a quantitative validation of the rooftop extraction results on our data, we manually created a set of ground-truth multi-polygon for 200 buildings scattered uniformly on the covered area. Then we evaluated the Intersection over Union (IoU) between the ground-truth and the input (non-aligned) and the output (aligned) multi-polygons.

In **Figure** 5, a bar plot showing the IoU score obtained for each considered building is reported. As can be seen, for almost all the test cases (only in four cases the input multi-polygons have higher IoU), the IoU increases using the output multi-polygons, confirming the effectiveness of the used approach. In average we obtain an IoU score of 0.7100 for the input multi-polygons, and 0.8854 for the output multi-polygons after align them using the deep network, with an increase of almost 17.5%.

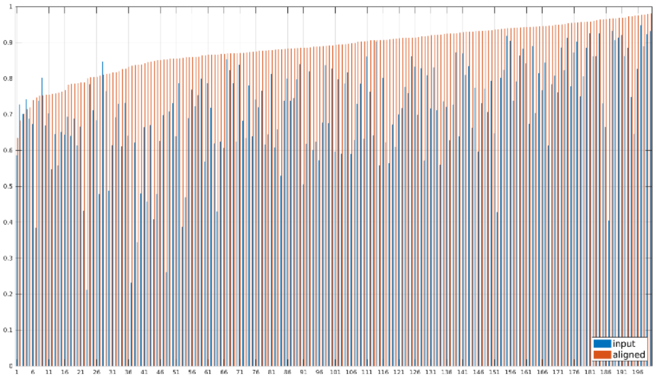


Figure 5: IoU scores for each of the 200 considered buildings. In blue the scores of the input (non-aligned) multi-polygons, in red the results on the output (aligned) multi-polygons. As can be seen, IoU increase for almost all the buildings on the aligned multi-polygons: only in four cases the input multi-polygons obtained better IoU. Note that results are ordered w.r.t. the aligned IoU scores for better readability.

VIII. Conclusions

In this paper, a system for implementing a 3D city model with photorealistic texture integrated into a Smart City framework has been presented. The proposed solution follows a deep learning approach based on U-Net to detect the rooftops from aerial images and align them with the 3D map buildings, which are obtained by extrusion from GeoJSON data. The solution is implemented in the open-source Snap4City platform as a multi-layer 3D map, which can be used by users as a widget on dashboards to visualize a full 3D city environment and a large variety of data, including IoT devices (city sensors and actuators, as well as private devices), POI, heatmaps, geometries and polylines related to cycling paths, bus routes, traffic flow etc. Specifically, users have the possibility to pick on map the single city elements and device markers and inspect their data and attributes. In this way, the proposed solution aims at providing an easy and smart navigation of the global digital twin of the city and the related data. The method employed for rooftop detection and alignment was validated against a set of 200 ground-truth multi-polygons extracted from aerial images of buildings uniformly scattered in the metropolitan area of Florence: after the alignment the, IoU score rises from 0.7370 to 0.8848, confirming the validity of the used approach. As a future work, an automatic procedure is going to be developed, in order to apply photorealistic texture also to building facades. Many other architecture details have been omitted for the lack of space such as the details regarding the content distribution, the production of facades, the exploitation of Lidar data.

Acknowledgment

Authors would like to thank the HeritData Interreg project. Snap4City (https://www.snap4city.org) is an open technology and research by DISIT Lab, University of Florence, Italy..

References

[1] K. Chaturvedi, A. Matheus, S. H. Nguyen and T. H. Kolbe, “Securing Spatial Data Infrastructures for Distributed Smart City applications and services,” Future Generation Computing Systems, vol. 101, pp. 723-736, 2019.

[2] N. Lafioune and M. St-Jacque, “Towards the creation of a searchable 3D smart city model,” Innovation & Management Review, vol. 17(3), pp. 285-305, 2020.

[3] G. Gröger and L. Plümer, “CityGML Interoperable semantic 3D city models,” ISPRS Journal of Photogrammetry and Remote Sensing, pp. 16-21, 2012.

[4] E. Shahat, C. T. Hyun and C. Yeom, “City Digital Twin Potentials: A Review and Research Agenda” MDPI, pp. 3, 2021.

[5] D. Jovanovic, S. Milovanov, I. Ruskovski, M. Govedarica, D. Sladic , A. Radulovic, and V. Pajic, “Building Virtual 3D City Model for Smart Cities Applications: A Case Study on Campus Area of the University of Novi Sad,” ISPRS International Journal of Geo-Information, pp. 16-21, 2020.

[6] Helsinki 3D city model. Available online: https://kartta.hel.fi/3d/#/

[7] ETH Zurich VarCity project. Available online: http://www.varcity.ethz.ch/

[8] Bonczak, B.; Kontokosta, C.E. «Large-scale parameterization of 3D building morphology in complex urban landscapes using aerial LiDAR and city administrative data.» Comput. Environ. Urban Syst. pp. 73, pp. 126–142, 2019.

[9] F. Xue, W. Lu, Z. Chen and C. J. Webster, “From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles,” ISPRS J. Photogramm. Remote Sens. pp. 167, pp. 418–431, 2020.

[10] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 117, pp. 11-28, 2016.

[11] J. A. Thompson, J. C. Bell and C. A. Butler, “Digital elevation model resolution: effects on terrain attribute calculation and quantitative soil-landscape modeling,” Geoderma, vol. 100, pp. 67-89, 2001.

[12] Y. Ye, J. Shan, L. Bruzzone and L. Shen, “Robust Registration of Multimodal Remote Sensing Images Based on Structural Similarity,” IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pp. 2941-2958, 2017.

[13] M. Izadi and P. Saeedi, “Automatic Building Detection in Aerial Images Using a Hierarchical Feature Based Image Segmentation,” in 2010 20th International Conference on Pattern Recognition, 2010.

[14] G. Mountrakis, J. Im and C. Ogole, “Support vector machines in remote sensing: A review,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 66, pp. 247-259, 2011.

[15] M. Belgiu and L. Drăguţ, “Random forest in remote sensing: A review of applications and future directions,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 114, pp. 24-31, 2016.

[16] H. Baluyan, B. Joshi, A. Hinai and W. Woon, “Novel Approach for Rooftop Detection Using Support Vector Machine,” ISRN Machine Vision, vol. 2013, p. 11, December 2013.

[17] M. Bosch, Z. Kurtz, S. Hagstrom and M. Brown, “A multiple view stereo benchmark for satellite imagery,” in 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2016.

[18] Y. Zhong, A. Ma, Y. soon Ong, Z. Zhu and L. Zhang, “Computational intelligence in optical remote sensing image processing,” Applied Soft Computing, vol. 64, pp. 75-93, 2018.

[19] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin and B. A. Johnson, “Deep learning in remote sensing applications: A meta-analysis and review,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 152, pp. 166-177, 2019.

[20] M. Chen and J. Li, “Deep convolutional neural network application on rooftop detection for aerial image,” ArXiv, vol. abs/1910.13509, 2019.

[21] K. He, G. Gkioxari, P. Dollar and R. Girshick, “Mask R-CNN,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

[22] R. Castello, A. Walch, R. Attias, R. Cadei, S. Jiang and J.-L. Scartezzini, “Quantification of the suitable rooftop area for solar panel installation from overhead imagery using Convolutional Neural Networks,” Journal of Physics: Conference Series, vol. 2042, p. 012002, November 2021.

[23] N. Girard, G. Charpiat and Y. Tarabalka, «Aligning and Updating Cadaster Maps with Aerial Images by Multi-task, Multi-resolution Deep Learning,» in ACCV, 2018.

[24] O. Ronneberger, P. Fischer and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Cham, 2015.

[25] A. Zampieri, G. Charpiat and Y. Tarabalka, “Coarse to fine non-rigid registration: a cain of scale-specific neural networks for multimodal image alignment with application to remote sensing,” ArXiv, vol. abs/1802.09816, 2018.

[26] S. Ruder, “An Overview of Multi-Task Learning in Deep Neural Networks,” ArXiv, vol. abs/1706.05098, 2017.

[27] Rotterdam 3D. Available online: https://www.3drotterdam.nl

[28] Berlin 3D, 3dcitydb. Available online: https://www.3dcitydb.org/3dcitydb-web-map/1.7/3dwebclient/index.html?title=Berlin_Demo&batchSize=1&latitude=52.517479728958044&longitude=13.411141287558161&height=534.3099172951087&heading=345.2992773976952&pitch=-44.26228062802528&roll=359.933888621294&layer_0=url%3Dhttps%253A%252F%252Fwww.3dcitydb.org%252F3dcitydb%252Ffileadmin%252Fmydata%252FBerlin_Demo%252FBerlin_Buildings_rgbTexture_ScaleFactor_0.3%252FBerlin_Buildings_rgbTexture_collada_MasterJSON.json%26name%3DBrlin_Buildings_rgbTexture%26active%3Dtrue%26spreadsheetUrl%3Dhttps%253A%252F%252Fwww.google.com%252Ffusiontables%252FDataSource%253Fdocid%253D19cuclDgIHMqrRQyBwLEztMLeGzP83IBWfEtKQA3B%2526pli%253D1%2523rows%253Aid%253D1%26cityobjectsJsonUrl%3D%26minLodPixels%3D100%26maxLodPixels%3D1.7976931348623157e%252B308%26maxSizeOfCachedTiles%3D200%26maxCountOfVisibleTiles%3D200

[29] Stockholm Opencities Planner. Available online: https://eu.opencitiesplanner.bentley.com/stockholm/stockholmvaxer

[30] Nesi, Paolo, et al. "An integrated smart city platform." Semanitic Keyword-based Search on Structured Data Sources. Springer, Cham, 2017.

[31] P. Bellini, F. Bugli, P. Nesi, G. Pantaleo, M. Paolucci, I. Zaza, "Data Flow Management and Visual Analytic for Big Data Smart City/IOT", 19th IEEE Int. Conf. on Scalable Computing and Communication, IEEE SCALCOM 2019, Leicester, UK https://www.slideshare.net/paolonesi/data-flow-management-and-visual-analytic-for-big-data-smart-cityiot

[32] E. Bellini, P. Bellini, D. Cenni, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, "An IoE and Big Multimedia Data approach for Urban Transport System resilience management in Smart City", Sensors, MDPI, 2021, https://www.mdpi.com/1424-8220/21/2/435/pdf

[33] C. Badii, P. Bellini, A. Difino, P. Nesi, "Sii-Mobility: an IOT/IOE architecture to enhance smart city services of mobility and transportation", Sensors, MDPI, 2019. https://doi.org/10.3390/s19010001 https://www.mdpi.com/1424-8220/19/1/1/pdf

Computer-Assisted Visual Reasoning for Territorial Intelligence

Robert Laurini
KSI, USA, and University of Lyon, France
Robert.Laurini@liris.cnrs.fr
ORCID : 0000-0003-0426-4030

Rosa Marina Donolo
Dept of Civil, Construction-Architectural and Environmental Engineering
University of L'Aquila, Italy
Rosamarina.donolo@univaq.it

Abstract— **In territorial intelligence, it is very interesting to provide computer-based tools to help reasoning especially in urban, regional and environmental planning. Traditionally, decision-makers use maps in their daily work, but they are limited in the expressive power to help reasoning, *i.e*, to assist them in deducing knowledge about salient problems, opportunities, and generating ideas and future projects. By means of visual analytics, and more specifically geovisualization, it seems possible. The scope of this paper is to rapidly analyze how painting (so visualization) has passed from representing objects as they are recognized to showing their relationships as a first step for reasoning. A similar study is made from conventional mapping to geovisualization, beyond traditional cartography, namely cartograms, chorems, datascapes, etc. as a way to base visual reasoning.**

Keywords—Visual Reasoning, Geovisualization, Territorial Intelligence, Smart Cities, Visual Analytics, Datascapes.

I. Introduction

For decision-makers in local authorities, it is important to capture and manage data, but overall information and knowledge in order to govern the territory under their jurisdiction. For that purpose, various software products have been created ranging for GIS systems, spatial analysis tools to, more recently, systems based on deep learning and knowledge management, i.e. systems with some reasoning capabilities, for instance in urban and regional planning. In geographic applications knowledge has no meaning in itself but derives its value from its use in practice. For a territory, knowledge corresponds to information potentially useful in order to make reasoning such as (Laurini et al. 2022):

- explaining and making understandable the dynamics of a territory as well as its interactions with other adjoining places in the same or neighboring countries.
- managing a territory by some local authorities, i.e. by means of some decision-support system, in the spirit of territorial intelligence;
- monitoring its daily development through feedbacks and adaptation;
- simulating the future, and design novel projects;
- orienting actions for the future.

In parallel, visual representation of territories has evolved from conventional cartography to geovisualization systems. According to MacEachren (2004) geovisualization (short for geographic visualization), also known as cartographic visualization, refers to a set of tools and techniques supporting the analysis of geospatial data through the use of interactive visualization. Like the related fields of scientific visualization and information visualization, geovisualization emphasizes knowledge construction over knowledge storage or information transmission. To do this, geovisualization communicates geospatial information in ways that, when combined with human understanding, allow for data exploration and decision-making processes (Jiang and Li, 2005; MacEachren, 2004).

What is visual reasoning? First, according to the Merriam-Webster dictionary, reasoning is defined *as the use of reason, and especially the drawing of inferences or conclusions through the use of reason*.

So, the research question addressed in this paper is to analyze how computer-based visualization can provide novel methods of reasoning about a territory. To help answer this question, we will study a few historical issues before the advent of computers, then examine what the possible solutions are existing now for visual reasoning.

II. Some Historical Landmarks

In this section, it seems interesting to study in what degree the visual representations were made in order to help reasoning. From a historical point of view, two directions will be detailed, namely painting and cartography.

A. Painting

The goal of this section is not to give a global history of painting but rather to examine a few ideas regarding the relationships between painting and the reality.

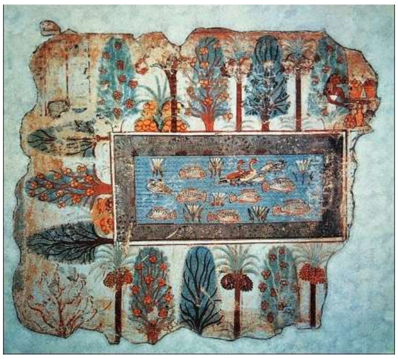


Fig. 1 Egyptians used to represent objects as they are recognized. Source https://www.britishmuseum.org/collection/object/Y_EA37983

Regarding prehistoric people, it seems that some paintings in cave have a sort of magic power to act on reality (Bégouen 1929). Later during the Egyptian period, the idea was to represent objects as they are recognized. For instance, in the famous Nebamon tumb, there is a painting (Figure 1) representing plants and animals in a garden: their flat representation allows anybody to recognize them without any problem. In the same idea, ask a child to represent a fish, s/he will not draw it from the top, from the bottom nor the face but rather from its side to easily recognize it.

Then, in painting, the dominant idea was rather to show some mythologic or religious paintings, sometimes far from reality. Later with the discovery of perspective, in 1435 Alberti wrote a treatise entitled De Pictura (On Painting) in which he outlined a process for creating an effective painting through the use of one-point perspective (Sinisgalli, 2011).

DOI reference number: 10.18293/DMSVIVA22-002

Now it was possible to represent objects as they are seen (see a painting in Figure 2 from Piero Perugino in 1481. The summum was reached by Leonardo da Vinci (1452-1519) in his Treatise on Painting (Trattato della pittura).

Fig. 2. Pietro Perugino's painting representing objects as they are seen. Source: https://www.analisidellopera.it/consegna-delle-chiavi-a-san-pietro-perugino/

Later during the early 20th-century art rebelled against the traditional understanding of painting. The painters focused on the relationship between the objects rather than on the traditional single-point perspective.

Fig. 3. Pablo Picasso's Gernica. Painting emphasizing relationship between objects. Source: https://www.museoreinasofia.es/en/collection/artwork/guernica

To conclude this section about painting, a sort of evolution schema can be outlined ranging from the visual representation of objects as they are recognized, as they are seen and finally the relationship between them which will be considered as a first step to reasoning.

B. Cartography of Small Territories

This is not the goal of this section to write a history of cartography and not to detail the ways of representing the whole earth, rather to give a few salient characteristics for maps made to reasoning.

The majority of maps are made to describe a territory, for instance in physical, economic and political geographies. Of course, even if they are limited to the description, they can assist any human to make reasoning, but their initial purpose was to explain.

Apparently, the first map was a slab discovered in Saint-Bélec, Brittany, from the Bronze Age as presented in Figure 4. It shows a region, presumably in 3D.

Several centuries after, there was the well-known Peutinger map. This is the only Roman world map known to have survived antiquity showing the Roman road network. Preserved in a single, medieval copy now housed in the Austrian National Library in Vienna, the map stretches from Britain in the west to India in the east, covering a series of 11 parchment rectangles. The idea behind this map is not to show the shape and locations of cities and rivers, but overall to help finding an itinerary from one city to another city. For instance, the Adriatic Sea is very narrow (at the top of Figure 5), whereas Carthage is depicted just below Rome!

Fig. 4. Saint-Bélec Slab showing a map of the Bronze Age. Source: https://allthatsinteresting.com/saint-belec-slab. Presenting things as they are located.

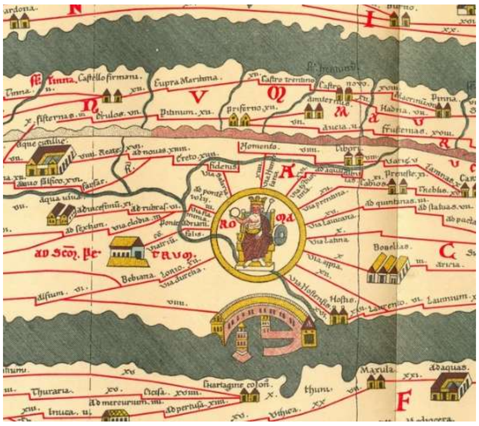


Fig. 5. An excerpt of the so-called Peutinger map centered in Rome, showing the Roman road network. Can be seen as a way of reasoning for finding itineraies. Source: https://www.onb.ac.at/

III. From Visual data mining to Geovisualization

In this section, we intend to clarify the differences between several notions such as visual data mining, visual analytics, visual reasoning and geovisualization

A. Visual Data Mining

According to Simoff (2020), visual data mining is the process of interaction and analytical reasoning with one or more visual representations of abstract data. At the difference with conventional data mining in which knowledge chunks are automatically collected often as patterns or associative rules, in visual data mining, the human interaction is the key: by observing particularities or regularities, someone can identify interesting issues, maybe leading to novel knowledge chunks.

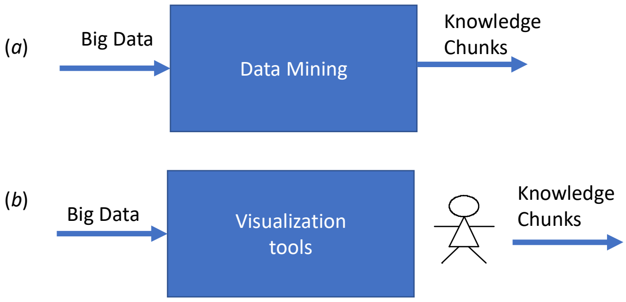


Fig. 6. Two ways of capturing knowledge. (a) via data mining. (b) via visualization tools (visual data mining).

In Figure 6, one can see the main difference between conventional data mining (*a*) in which knowledge chunks are identified by a computer whereas in visual data mining (*b*), knowledge chunks are identified by a human.

B. *Visual Analytics*

Wong and Thomas (2004) gave the following definition: visual analytics is an outgrowth of the fields of information visualization and scientific visualization that focuses on analytical reasoning facilitated by interactive visual interfaces. In Figure 7, Keim et al. (2010) depict a visual analytics workflow as a methodology to produce knowledge through visualization.

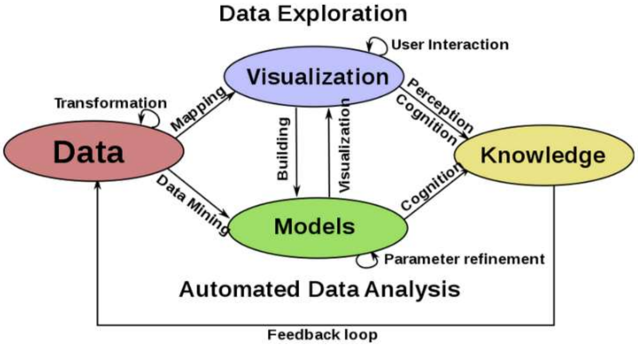


Fig. 7. Visual analytics workflow. From Keim et al. (2010).

According to Thomas-Cook (2005) here are a few recommendations in research about visual analytics:

- build upon theoretical foundations of reasoning, sense-making, cognition, and perception to create visually enabled tools to support collaborative analytic reasoning about complex and dynamic problems;
- create a science of visual representations based on cognitive and perceptual principles that can be deployed through engineered, reusable components;
- develop a new science of interactions that supports the analytical reasoning process;
- create methods to synthesize information of different types and from different sources into a unified data representation so that analysts, first responders, and border personnel may focus on the meaning of the data.

C. *Visual Reasoning*

Emily Daw (2022) defines visual reasoning as the process of analyzing visual information and being able to solve problems based upon it. In other words, visual reasoning produces knowledge. But what are the characteristics of this knowledge? For humans, overall knowledge is verbal but multimedia knowledge is also important: think about music, images, videos, gastronomy, domains for which sometimes it is difficult to transmit knowledge with words.

A good example of visual reasoning is crime board (Figure 8) which can be seen in practical all police series and detective shows: in this board are pinned suspects' photos, crime location, relationships between them, pieces of evidence, etc. As soon as new information is discovered, this is put on the board. By looking at it, detectives formulate assumptions. Often the solution comes from a missing relation.

In this paper, we only try to examine how knowledge can derive from figures, drawings, schemata and especially from information mapping, i.e. coming from visual analytics, and geovisualization.

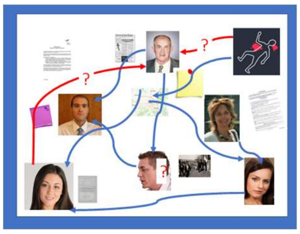


Fig. 8. Example of a detective crime board. By visual reasoning,the solution of the crime is found.

D. *Definition of Geovisualization*

According to MacEachren and Kraak (1997), Geovisualization can be defined as a set of tools and techniques to support geospatial data analysis through the use of interactive visualization. Like the related fields of scientific visualization and information visualization, geovisualization emphasizes information transmission. Geovisualization communicates geospatial information in ways that, combined with human understanding, allow data exploration and decision-making processes. Beyond cartography whose goal is representing territory with fidelity (usually physical or topographical), geovisualization tries to help highlight the more important issues.

To summarize, geovisualization is an interesting and useful field of research for different reasons:

- it can reduce the time to search information, and support decision-making;
- it can enhance the recognition of patterns, relations, trends and critical points etc.;
- it can give a global vision of a situation, a phenomenon, etc.;
- it enables the use of human visual memory and the capability of perceptual processing of data;
- it permits a better interaction between user and the information system;
- and it can possibly lead to the discovery of new bunches of knowledge.

IV. GEOVISUALIZATION FOR REASONING

According to Lacoste (1976), in his provocative book, explained that geography was a form of strategic and political knowledge, central to the military strategy and the exercise of political power. In other works, geography help reasoning for war. But the role of geography is more than that since it permits reasoning in other domains such as urban, regional and environmental planning. For instance, geographic reasoning is useful for the following issues (Laurini 2020):

- Where to put a new airport, a new hospital, a new stadium, a new recreational park, etc.?
- Is this new construction project compliant with planning rules?
- What is the best mode or the best way to get from *A* to *B*?
- How to organize a plan for green spaces in a city?
- How to reorganize common transportation?
- How to transform slum sectors into more modern houses?
- What could be the cost of a projected operation?

A. *Generalities about Geovisualization*

Anyhow, perhaps one of the first geovisualization display was made by Minard regarding the march of Napoleon against Russia as depicted in Figure 9 in which the size of the line is proportional to the number of soldiers, yellow when going and black when returning. Due to bad temperatures, he lost more or less 2/3 of his army when marching; during the battle few soldiers died, and in the way back very few soldiers returned home.

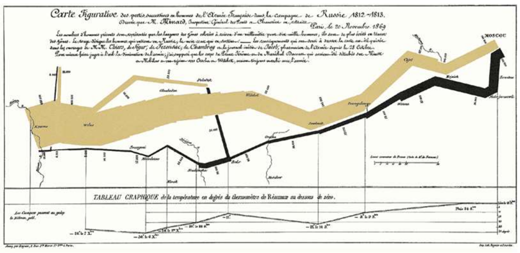

Carte Figurative des pertes successives en hommes de l'Armée Française dans la Campagne de Russie 1812-1813.
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.

Fig. 9. March of Napoleon againt Russia by Minard. Source: https://www.edwardtufte.com/tufte/minard. It is considered as a first geovisual representation.

Among geovisualization methods, let us rapidly present cartograms, chorems, datascapes and 3D representations. The first two can be seen as 2D extensions of thematic cartography whereas datascapes be considered as 2½ D.

B. Cartograms

According to Grover (2014) cartograms are a kind of maps which take some measurable variable: total population, age of inhabitants, electoral votes, GDP, etc., and then manipulate a place's area to be sized accordingly. The produced cartogram can really look quite different from the maps of cities, states, countries, and the world that are more recognizable. It all depends on how a cartographer needs or wants to display the information. An example is given Figure 10 showing GDP wealth in 2018; look at Russia, China and Africa for distortions. There are various forms of cartograms (Field 2017):

- **non-contiguous cartogram:** adjacencies are compromised as areas shrink or grow; individual area shapes are kept but they become detached from the overall map;
- **contiguous cartogram:** adjacencies are maintained but shape is distorted to accommodate the mapped variable
- **graphical cartogram:** maintains neither shape, topology or location; instead using non-overlapping geometric shapes (e.g. circles or squares) to represent the mapped variable (see example Figure 11);
- **gridded cartogram**: uses repeating shapes of the same or different size to create a tessellated representation of the mapped variable;
- **topology:** non-metric spatial relationships that are preserved under continuous transformation *e.g.* adjacency.

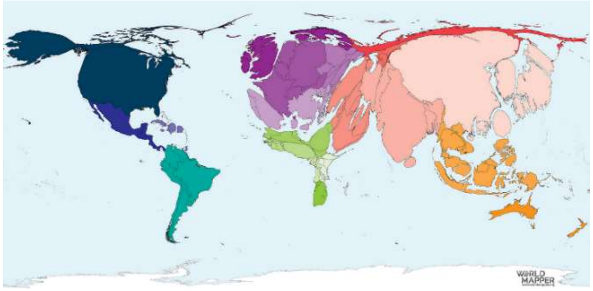


Fig. 10. Example of a cartogram emphasizing Gross Domestic Product wealth in 2018. Source https://worldmapper.org/maps/gdp-2018/

Cartograms, by adapting shapes in accordance a very well-defined variable, permit reasoning which were not possible with traditional.

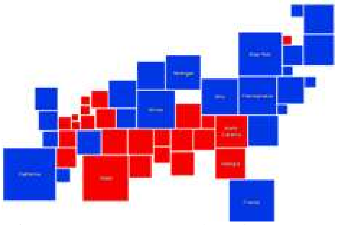


Fig. 11. Example of a cartogram presenting the U.S. Presidential election results as a cartogram based on squares. Source: https://gistbok.ucgis.org/bok-topics/cartograms.

C. Chorems

Chorems were created in 1980 by Pr. Roger Brunet (1980), a French geographer as a schematic representation of a territory. This word comes from the Greek χώρα which means space, territory. It is not a raw simplification of the reality, but rather aims at representing the whole complexity with simple geometric shapes. Even if it looks a simplification, the chorem tries to represent the structure and the evolution of a territory with a rigorous manner. The basis of a chorem is in general a geometric shape in which some other shapes symbolize the past and current mechanisms. Brunet has proposed a table of 28 elementary chorems, each of them representing an elementary spatial configuration, and so allowing them to represent various spatial phenomena at different scales.

According to Brunet (1980), chorems are a tool among other to model the reality, but it is a very precious tool not only as a visual system, but also as a spatial analysis too. Considering Mexico, Figure 12 presents both a traditional (physical) map and a chorematic map in which the salient issues are considered (Lopez et al. 2009).

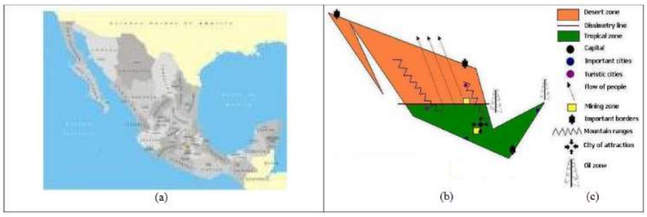


Fig. 12 Example of chorem; (*a*) A traditional map, (*b*) a chorematic map of Mexico and (*c*) its legend. (Lopez et al. 2009).

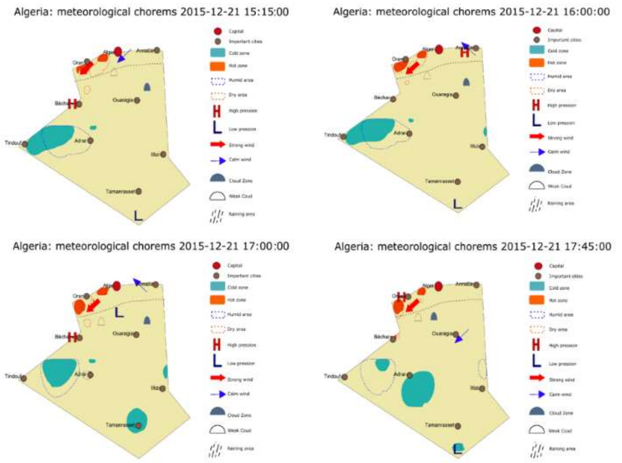

Algeria: meteorological chorems 2015-12-21 15:15:00
Algeria: meteorological chorems 2015-12-21 16:00:00
Algeria: meteorological chorems 2015-12-21 17:00:00
Algeria: meteorological chorems 2015-12-21 17:45:00
Capital
Important cities
Cold zone
Hot zone
Humid area
Dry area
High pressure
Low pressure
Strong wind
Calm wind
Cloud Zone
Weak Cloud
Raining area
Tindouf
Adrar
Bechar
Oran
Alger
Annaba
Ouargla
Illizi
Tamanrasset
H
L

Fig. 13 Example of animated chorem showing weather evolution at different dates in Algeria (Bouattou et al. (2017).

In Bouattou et al. (2017) an experimentation of chorem generation in real time is presented with an application in meteorology in Algeria (Figure 13) in which animated chorems could the possible base for real time reasoning.

To conclude this rapid presentation about chorems, let us say that they can be seen as a sort of generalization, both geometrically and semantically.

Those salient issues can be extracted by data mining and then visualized; thanks to the chorematic presentation, new knowledge bundles can be discovered.

D. 3D Representations and Datascapes

An innovative data driven graphic approach to model environmental, territorial and urban systems is the representation of natural and anthropic phenomena as "datascapes", literally "data landscapes".

This approach integrates the approaches that describe the elements of the "physical and real" landscape systems, that can traditionally be represented by means of the 2D GIS cartography, and with the datascape representation that allows also the description of the "visible and invisible" phenomena and that spatially represents them using the third dimension, in a "virtual landscape of data" (datascape) in 3D.

Nadia Amoroso (2010), expert in Data Visualization, defines datascapes as "a visual representation of all the quantifiable forces that influence a system", and also as "digital landscapes". See Figures 14, 15 and 16.

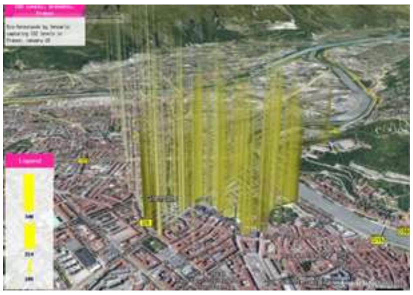


Fig. 14. Datascape in urban area (by N. Amoroso), data elaboration with the software DataAppeal; presence of CO2 in Grenoble, France. Source: https://archinect.com

The datascape representation is particularly suitable to model different kinds of territorial systems: from simple structural systems that only describe the physical structure of a system, to more complex systems such as urban systems and ecological networks, in which many non-structural and non-visible aspects are added. In fact, in the datascape approach can describe all the different elements and phenomena that are present simultaneously and in the same place that can be quantitatively described and that, to be described, require a large amount of data.

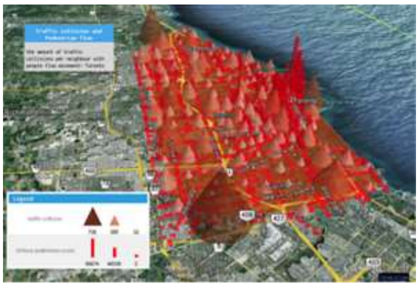


Fig. 15. Datascape in urban area (by N. Amoroso), data elaboration with the software DataAppeal; Traffic accidents and pedestrian traffic in Toronto. Source: https://archinect.com

In the example of the representation of ecological networks, there is a combination of the biodiversity protection aim combined with the aim of implementing eco-systemic services in the territory; therefore, new elements are taken into consideration and integrated, which increase the degree of complexity of the system-network and of the phenomena associated with it. Furthermore, the "datascape modeling" is suitable to represent variables of non-visible urban and extra-urban phenomena (temperature, pollution, etc.).

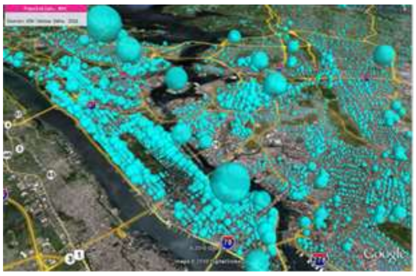


Fig. 16. Datascape in urban area (by N. Amoroso), data elaboration with the software DataAppeal; Population density in NY. Source: https://archinect.com

Differences between cartograms, chorems and datascapes

The main difference between cartograms, chorems and datascapes is that cartograms and chorems deform the real cartographic representation, instead the datascape representation displays simultaneously the real cartographic representation and the virtual representation of the variable that we need to represent (Donolo-Laurini 2015).

The cartographic representations of territorial data can be very complex, since they present different levels of information at the same time and in a multi-dimensional context; one of the main advantages of datascape modeling is given by the potential of representation, for both spatial and temporal dimensions, of the numerous geo-spatial elements and associated phenomena that can be present simultaneously in a territory; in particular, it should be noted that:

a) **For the spatial dimension:** In the territorial system, in addition to the physical and visible elements in the three dimensions of the "physical" space (*x*, *y*, *z*), there are simultaneously visible and representable phenomena (e.g. presence of a population of wild animals present in a specific area, in a given period of time) and non-visible phenomena (*e.g.* presence of atmospheric or aquatic pollutants, or variation of the average temperature of the habitats, etc.), which however can be visualized and represented using the three dimensions of a “metaphorical or virtual” graphic space (x_1, y_1, z_1) which can be superimposed on physical space (*x*, *y*, *z*), since it can be translated along the *z* axis.

b) **For the temporal dimension:** In the territorial system different phenomena occur simultaneously, and the "metaphorical space" helps to visualize them, both because it is replicable and superimposable on the physical space, and because it is a space in which the quantities that vary over time can be represented dynamically in real time.

Advantages of the Datascape modeling

The advantage of having an additional virtual space for displaying data and indicators brings with it other positive aspects:

1. Since the variable *z* of the "metaphorical space in 3D (*x, y, z*)" can assume both positive and negative values, it is also possible to use the representation space underlying the surface identified by the values with *z* = 0, consequently it increases the space that can be used to graphically represent the phenomena in a single "view" of digital cartography.

2. As already known for the representations of ecological networks in 2D with GIS tools, an advantage of the graphic representation of datascape in 3D is the possibility to customize and optimize the use of graphic variables (color, shape, size, etc.), also called visual variables, and their combinations. In fact, as the number of variables available increases (considering that the variables of the new "metaphorical space" (x_1, y_1, z_1), and the time variable t, are added to the spatial variables (x, y, z), of the territory under examination), the number of visual variables that can be associated to the spatial variables (x, y, z, x_1, y_1, z_1, and t) also increases and therefore also increases the possible combinations between the visual variables and the possible ways of representing the same phenomenon. Visual variables have been defined as "a way to modify graphic signs" (Pumain et al., 1989). In particular, datascape modeling presents an additional visual variable compared to the traditional 2D GIS representation which is possible exactly the third additional dimension: "the perspective". The main visual variables whose most effective use was analyzed to highlight qualitative or quantitative differences between the graphic objects represented by Jacques Bertin (1967). It should be noted that other researchers have subsequently discussed and expanded the systematization of Bertin's visual variables, and their optimal use based on the properties of the data to be represented.

3. As a third positive aspect, it should be noted that the datascapes can be processed online with specific software that uses the interactive 3D map base of Google-Earth on the Web, which can be rotated and zoomed; with it you can therefore rotate and zoom all the datascape representations superimposed on it: so the datascape not only are dynamic, but also interactive and allow you to explore the map from different perspectives, in order to bring out relationships between phenomena, trends, hidden criticalities, etc.

4. Another advantage that we want to highlight is given by the fact that this datascape modeling not only allows to manage a large amount of data, but also different data configurations (datascape); in fact, through appropriate Dashboards it is possible the management and modification in real time of both single datascapes and multiple datascapes combined and correlated with each other: although a configuration of natural and anthropic phenomena can be temporally stabilized, at times, it is sufficient for a single phenomenon to undergo a modification, to modify a whole series of phenomena; so it will not be sufficient to analyze a single reality, but a set of different possible realities. Even the digital environments of datascapes are multiple, fluid and are able to represent and monitor different configurations of phenomena; As an example, different datascapes can be used to represent ecological networks belonging to different animal populations.

5. It is possible to elaborate datascapes not only of the phenomena detected at a given moment, but also of possible future scenarios, which can also be compared with each other. could also compare datascapes of phenomena belonging to different categories, for example by representing all datascapes deriving from anthropic phenomena under the $z = 0$ plan and all datascapes of natural phenomena above the $z = 0$ plan.

6. Working on datascapes on a large area scale: as the complexity of the ecological network increases with the increasing size of the territorial area concerned, with the datascape modeling it is possible to work at different scales, which can also be compared.

7. With the datascape modelling it is possible to customize and optimize of the graphic symbology of the single datascape and therefore the management of the visual variables.

8. With the datascape modelling it is possible the identification of hierarchies between datascapes, which can be highlighted.

Critical aspects of the Datascape modeling

Some critical aspects to consider in the analysis of territorial systems / networks using the datascape approach, are:

1. The first aspect to consider is the tendency to abuse the technological potential and to represent too many territorial and visual variables at the same time in a single "view / display", decreasing the readability of the maps by both expert and non-expert users; there are studies in the psycho-cognitive literature that aim to evaluate how many territorial and visual variables are able to simultaneously process human visual and cognitive capacity. In particular, it emerged that some combinations of territorial variables and visual variables are more effective than others for representing certain quantities or phenomena.

2. A second aspect to note is that, as previously mentioned, by having the information layers necessary to elaborate the ecological network of a territory, not one, but more datascapes will be produced, and therefore a necessary development of the "datascape approach" it will concern the modification, analysis, comparison and display of multiple datascapes at the same time; although this is generally possible through a Dashboard, which is used to manage the different datascapes, it remains a critical aspect, because it requires use by experts at least in the design phase. A Dashboard, however, can be made available online, and can allow even the non-expert user to interact on the Web, and can make changes and views in real time shared between multiple users. Figure 17 shows an example of the interface of the DataAppeal software Dashboard implemented by Nadia Amoroso.

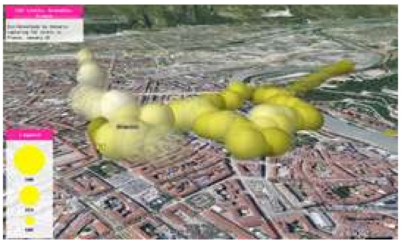


Fig. 17. Example of datascape representation on the Web, which represents, with the Dashboard for customization (to make graphic and analytical changes, including interactive online and in real time). Source: https://archinect.com

1. A third critical aspect concerns the poor readability of any graphic symbols / labels: also, in the case of datascapes, as for GIS maps in 2D, the use of graphic symbols superimposed on the cartography, adds information to the representation, but in the case of datascapes the perspective view in 3D could in some cases prevent them from being read.

E. Other Examples of 3D Representations

In Figure 18, is shown a comparison between the representation of the variable "diffusion of a product on the market": on the left the "traditional" GIS approach in 2D, on the right the "datascape" approach in 3D.

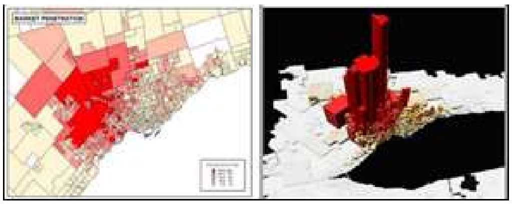


Fig. 18. Representation of the variable "diffusion of a product on the market" with two different approaches. Source: https://archinect.com

Visual Reasoning with Datascapes

For territorial intelligence, datascapes can assist visual reasoning in several directions:

- Detection of outliers: by observing a place where data are strange, a quality control action must be launched;
- Detection of a novel problem either social or environmental;
- Detection of new patterns, for instance polarization around the CBD or along a traffic route;
- Detection of spatial correlation by comparing two datascape of the same place;
- Confirmation of already-known information.
- Etc.

IV. Conclusion

The scope of this paper was to examine how computer-assisted visual reasoning can help territorial intelligence. Starting from the evolution of painting and cartography, we have tried to show how geovisualization can highlight reasoning. Of course, other geovisualization methods may exist, but the general mechanisms are given Figure 19: sensors about a city send data to a geovisualization tools; the results are examined by an expert who can capture some knowledge chunks; those chunks added to already-known chunks are sent to a reasoning tool which infers suggestions; those solutions are studied by local decision-makers to launch actions plans or actions.

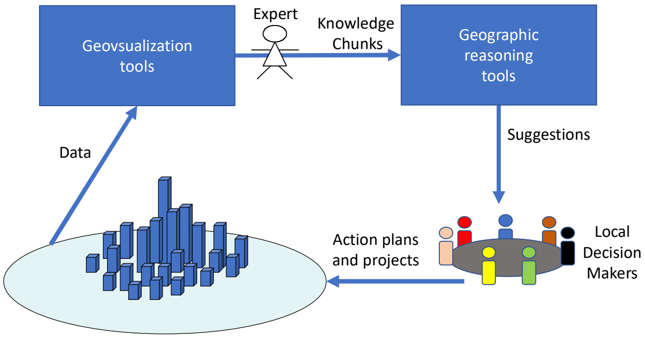


Fig. 19. Structure of visual reasoning for territorial intelligence.

Anyhow, this statement raises three kinds of questions:

a – Are they existing other methods of visualization more adapted to derive visual knowledge?

b – What are the main characteristics of visual knowledge chunks and bundles issued from geovisualization?

c – How to formalize them to be used in inference engines?

References

[1] Amoroso N., 2010, The Exposed City, Mapping the Urban Invisibles, Routledge.

[2] Bégouen, C. (1929). The Magic Origin of Prehistoric Art. Antiquity, 3(9), 5-19. doi:10.1017/S0003598X00002933

[3] Bertin J., (1967) Sémiologie graphique, Les diagrammes, les réseaux et les cartes, Mouton, Gauthier-Villars, Paris.

[4] Bouattou Z., Laurini R., Belbachir H. (2017) Animated Chorem-based Summaries of Geographic Data Streams from Sensors in Real Time. Journal of Visual Languages & Computing. April 2017.

[5] Brunet R. (1980) "La composition des modèles dans l'analyse spatiale", in L'Espace géographique, no. 4, 1980.

[6] Daw E. (2022) What Is Visual Reasoning? https://www.infobloom.com/what-is-visual-reasoning.htm

[7] Donolo R.M., Laurini R., (2015) Evaluation and improvement of smart representations of urban data for developing smart cities, Proceedings 18th AGILE Conference on Geographic Information Science, Geographic Information Science as an enabler of smarter cities and communities, Lisboa.

[8] Field, K. (2017). Cartograms. The Geographic Information Science & Technology Body of Knowledge (3rd Quarter 2017 Edition), John P. Wilson (ed). DOI: 10.22224/gistbok/2017.3.8

[9] Grover D. (2014) What is a Cartogram? https://populationeducation.org/what-cartogram/

[10] Keim D.A, Kohlhammer J., Ellis G.P., Mansmann F. (2010) Mastering The Information Age - Solving Problems with Visual Analytics. Eurographics, 2010.

[11] Jiang, B., and Li, Z. 2005. Editorial: Geovisualization: Design, Enhanced Visual Tools and Applications. The Cartographic Journal, 42(1), pp. 3–4.

[12] Lacoste Y. (1976) La Géographie ça sert d'abord à faire la guerre. Paris : La Découverte. ISBN 2-7071-0815-4

[13] Laurini R. (2020) A primer of knowledge management for smart city governance. Land Use Policy, 2020, 104832, ISSN 0264-8377, https://doi.org/10.1016/j.landusepol.2020.104832.

[14] Laurini R., Nijkamp P., Bordogna G., Kourtit K., Duchateau F., Rinaldi A., Bouzouina L., Mehaffy M.E., Anthony B. (2022) "Regional Knowledge Management and Sustainable Regional Development: In Quest of a Research and Knowledge Agenda". In Laurini R., Bouzouina L., Kourtit K., Nijkamp P. (eds (2022)) "Knowledge Management for Regional Policymaking" Springer Verlag. On press.

[15] Lopez K., Laurini R., Del Fatto V., Sol D., Loreto R., Sebillo M., Vitiello G. (2009) "Visualizing Geographical Analysis Results From Spatial Databases Based on the Chorems". Proceedings of the 2009 International Conference on Modeling, Simulation and Visualization Methods (MSV'09: July 13-16, 2009, USA).

[16] MacEachren, A.M., (2004) How maps work: representation, visualization, and design. Guilford Press.

[17] MacEachren A.M., Kraak M.J., (1997) "Exploratory cartographic visualization: advancing the agenda", Computers & Geosciences, vol. 23, no. 4, pp. 335–343.

[18] Pumain D., Sanders l., Saint-Julien T. (1989) Villes et auto-organisation, Economica.

[19] Simoff S.J., Böhlen M.H., Mazeika A. (2020) "Visual Data Mining: An Introduction and Overview". In Visual Data Mining pp 1-12. Springer Lecture Notes in Computer Science book series (LNCS, volume 4404)

[20] Sinisgalli R. 2011. On Painting: A New Translation and Critical Edition. Original in Latin and Italian "De Pictura" written by leon Battista Alberti in 1435. Cambridge University Press, 2011. xvi +214 pp.. ISBN: 978–1–107–00062–9.

[21] Thomas J.J., Cook K.A. (2005). Illuminating the Path R&D Agenda for Visual Analytics, National Visualization and Analytics Center (NVAC), US Dept of Homeland Security, available at: http://nvac.pnl.gov/ agenda.stm, 2005.

[22] Wong P.C., J. Thomas J. (2004). "Visual Analytics". in: IEEE Computer Graphics and Applications, Volume 24, Issue 5, Sept.-Oct. 2004 pp. 20–21.

Graphical Animations of an Autonomous Vehicle Merging Protocol

Dang Duy Bui, Minxuan Liu, Kazuhiro Ogata
School of Information Science
Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{*bddang,liuminxuan,ogata*}*@jaist.ac.jp*

***Abstract*—State machine graphical animation (SMGA) is a tool that can help humans conjecture characteristics of protocols by observing graphical animations of state machines formalizing the protocols. SMGA requires a state picture template that is designed by users. Designing the state picture template is a core task of SMGA. In the present paper, we graphically animate an autonomous vehicle merging protocol in which we show some aspects affecting the state picture template, such as some properties of the protocol and Gestalt principles. Based on our design, we conjecture some characteristics of the protocol by observing graphical animations and also confirm the characteristics with Maude. Finally, we summarize our experiences as a procedure on how to design a state picture template and the procedure can be used when making a state picture template from scratch.**

***Keywords*-autonomous vehicle merging protocol; graphical animations; Maude; SMGA; state picture template.**

I. Introduction

State Machine Graphical Animation [1] is a visualization tool that graphically animates state machines formalizing protocols. The main purpose of the tool is to assist humans to conjecture characteristics of the protocol where such characteristics can be used as lemma candidates in theorem proving. Some studies [2], [3], [4], [5] have been conducted to demonstrate the usefulness of SMGA. One of them [4] has shown that the state picture template (an input of SMGA) is a crucial part of SMGA. In the present paper, we mainly provide some aspects that affect the design of the state picture template by describing in detail how to make graphical animations of a case study from scratch.

Frank et al. [6] have proposed a method to visualize state transition systems. They aim to let users observe the global properties of protocols by visualizing (large) state spaces as a backbone tree with the cone tree concept. The result shows that users can find some global properties of protocols, such as obtaining some clusters containing states that do not return to initial nodes after starting some executions. Bui et al. [5] have used SMGA to graphically animate the intersection traffic control distributed mutual exclusion protocol or the LJPL protocol [8]. In the LJPL protocol, there are eight lanes where vehicles in conflicted lanes cannot enter an intersection at the same time. The authors have made the state picture template of the LJPL protocol and revised SMGA so that SMGA can visualize queues formalizing lanes which elements in the queues are visualized by other variables. Some characteristics are conjectured via observing graphical animations based on their state picture template.

In near future, cars could become self-driving vehicles. Many techniques/protocols have been proposed to control such vehicles. Intuitively, autonomous cars must work well in many kinds of situations, such as moving in an intersection. Autonomous vehicle merging protocol [9] or the AR protocol is one possible way to control autonomous vehicles to avoid crashing each other at a merge point that is an intersection of two lanes. Liu et al. [10] have revised and formally specified the AR protocol in Maude. In the present paper, we use the AR protocol as a candidate for SMGA.

To make graphical animations of the AR protocol, we first need to design the state picture template. Two lanes are important in the AR protocol and they are (partly) specified by two queues in [10]. In the present paper, we combine the visualization technique [5] and some existing visualization techniques to design the queues in the AR protocol. When making the state picture template, we use some tips [4] to create some first drafts. When observing the drafts, we redesign some elements using Gestalt principles [11], [12] and point out some factors affecting the designs of the elements in the protocol, such as colors via the similarity law in the Gestalt principles.

We provide a state picture template that can be used in case the number of vehicles participating in the protocol is up to a fixed number. Therefore, users do not need to redesign the state picture template when the number of vehicles participating in the protocol is less or equal to the fixed number. Some characteristics of the AR protocol are conjectured by observing graphical animations based on our design, and the characteristics are also confirmed with Maude. Finally, we summarize some lessons learned a procedure on how to design a state picture template. The procedure can be used with useful factors (such as colors) when starting making a state picture template.

The rest of the paper is structured as follows. We introduce

DOI reference number: 10.18293/DMSVIVA22-009

the AR protocol and its specification in Sect. II. In Sect. III, we describe ideas in detail on how to design the state picture template of the AR protocol from scratch and give some factors affecting the state picture template. In Sect. IV, we guess some characteristics of the AR protocol, confirm them with Maude and summarize our experiences as a procedure when making the state picture templates from scratch. Finally, we conclude the present paper in Sect. V.

We suppose that readers are familiar with state machines and some existing techniques of SMGA.

II. An Autonomous Vehicle Merging Protocol

A. Description

An autonomous vehicle merging protocol (or the AR protocol) has been proposed by S. Aoki and K. Rajkumar [9]. In the protocol, there are two lanes called through and non-through lanes. The intersection of two lanes is called a merge point shown in Fig. 1. Vehicles on both lanes are supposed that they can run toward the merge point and in one direction only. At the merge point, the AR protocol controls that vehicles never collide with each other.

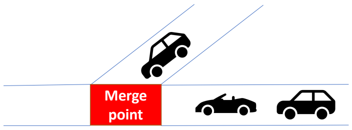


Figure 1. A merge point

In the protocol, there are two versions corresponding to two traffic environments: (1) only autonomous vehicles on the traffic (homogeneous traffic) and (2) autonomous vehicles and human-driven vehicles on the traffic (heterogeneous traffic). Liu et al. [10] have revised the first version so that this revised version does not rely on any real-time information, such as speed of vehicles running on both lanes. In the present paper, we use this revised version for our purpose. Therefore, in the rest of the paper, the AR protocol is referred to the revised version [10]. There are two modes in the AR protocol: prioritized and fair. In the prioritized mode, vehicles in the non-through lane (or non-through lane vehicles) cannot enter the merge point if some vehicles in the through lane (or through lane vehicles) are approaching the merge point. Basically, there are three cases:

1) If some through lane vehicles are approaching the merge point and there is not enough space between any of two adjacent vehicles, then non-through lane vehicles must stop before the merge point until all through lane vehicles have passed through the merge point. The following figure is an example of this case:
2) If no through-lane vehicle is approaching the merge point, non-through lane vehicles can enter the merge point.
3) If there is enough space between any two adjacent through lane vehicles, then non-through lane vehicles can use the space to enter the merge point. The following figure is an example of this case:

When the traffic of the through lane becomes congested, the prioritized mode changes to the fair mode. In the fair mode, through lane vehicles and non-through lane vehicles can enter the merge point alternatively. If the traffic of the through lane becomes less congested, the mode changes back to the prioritized mode.

In the AR protocol, there are five statuses for each vehicle: *running*, *approaching*, *stopped*, *crossing*, and *crossed*. Those statuses are summarized as follows:

- *running*: when a vehicle is far away from the merge point, its status is *running*.
- *approaching*: when a vehicle gets close to the merge point, its status changes from *running* to *approaching*.
- *stopped*: when a vehicle meets some conditions to stop before the merge point, its status changes from *approaching* to *stopped*.
- *crossing*: if a vehicle can enter the merge point, its status changes from *approaching* or *stopped* to *crossing*.
- *crossed*: when a vehicle has passed the merge point, its status changes from *crossing* to *crossed*.

In the protocol, for through lane vehicles, there are some situations or conditions to control their statuses. From *running* to *approaching*, there is one case that through lane vehicles get close to the merge point. From *approaching* to *stopped*, there are two cases as follows:

- There is a vehicle in the merge point, then a vehicle at the top of the through lane stops before the merge point.
- When the protocol is in the fair mode and in the non-through lane's turn, then a vehicle at the top of the through lane stops before the merge point.

From *approaching* to *crossing*, there are two cases as follows:

- When the protocol is in prioritized mode and there is no vehicle in the merge point, then a vehicle at the top of the through lane enters the merge point.
- When the protocol is in fair mode, the current turn is the through lane and there is no vehicle in the merge point, then a vehicle at the top of the through lane enters the merge point.

From *stopped* to *crossing*, there are three case in which two cases are similar or the same to two cases when *approaching* to *crossing*. The left case can be described as follows:

- When the protocol is in fair mode, the current turn is the non-through lane's, there is no vehicle in the merge point and there is no non-through lane vehicle in the queue, then a vehicle at the top of the through lane enters the merge point.

From *crossing* to *crossed*, there is only one case where through vehicles have passed the merge point. For non-through lane vehicles and others, please refer to [10] in detail.

B. Formal Specification of the AR Protocol in Maude

As mentioned, Liu et al. [10] have formally specified the AR protocol in Maude. First, two lanes are specified as two queues of vehicles. In the through lane, vehicles are specified as real vehicles and dummy vehicles (or spaces). In the non-through lane, vehicles are specified as real vehicles only. A space contains three kinds of statuses named *unspace*, *space*, and *yield* referring to the space that is not in the queue, is in the queue, and has just been out of the queue, respectively. All observable components are used to formalize the AR protocol as follows:

- `(v[`*ID*`]:` L, *VS*`)` - represents the *ID*'s vehicle that contains the vehicle lane information L (such as the through and the non-through lane) and the status *VS* of this vehicle. *ID* can be $v(i)$ or $dv(i)$ for real vehicles or spaces (dummy vehicles), respectively. L can be *through* or *nonThrough*. For real vehicles, *VS* can be *running*, *approaching*, *stopped*, *crossing*, or *crossed*; initially, *VS* is *running*. For dummy vehicles, *VS* can be *uspace*, *space*, or *yield*; initially, *VS* is *unspace*.
- `(lane[`*L1*`]:` Q`)` - represents the lane in the protocol where *L1* can be *through* or *nonThrough* corresponding to the through lane or the non-through lane, respectively. Q is a queue of vehicle *IDs*. If *L1* is *through*, the queue can contain spaces; otherwise, the queue contains real vehicles only. Initially, Q is *empq*.
- `(crossing?:` B`)` - represents whether a vehicle is in the merge point. If so, B is *true*; otherwise it is *false*. Initially, B is *false*.
- `(mode:` M`)` - represents the mode in the AR protocol. M can be *prioritized* or *fair* corresponding prioritized and fair mode, respectively. Initially, M is *prioritized*.
- `(turn:` *L2*`)` - represents the turn when the system is in the fair mode. *L2* can be *through* or *nonThrough* referring to the through lane vehicle turn and the non-through lane vehicle turn. Initially, *L2* is *through*.
- `(#uvcs:` N`)` - represents the number of vehicles that have not yet passed to the merge point. N is a natural number. When N is 0, all vehicles have crossed the merge point. Initially, N is the number of the vehicles concerned.
- `(gstat:` F`)` - indicate that all vehicles have passed the merge point. F can be *fin* or *nFin* meaning that all vehicles have passed or have not yet passed the merge point, respectively Initially, F is *nFin*. Note, this observable component and `#uvcs` are used to stop the protocol.

When there are two vehicles participating in the non-through lane, and four vehicles and two spaces participating in the through lane, an initial state can be expressed as follows:

```
(gstat: nFin) (#ucvs: 6) (crossing?: false)
(mode: prioritized) (turn: through)
(lane[through]: empq) (lane[nonThrough]: empq)
(v[v(0)]: through,running) (v[v(1)]: through,running)
(v[v(2)]: through,running) (v[v(3)]: through,running)
(v[v(4)]: nonThrough,running) (v[dv(0)]: through,unspace)
(v[v(5)]: nonThrough,running) (v[dv(1)]: through,unspace)
```

There are several rewrite rules that express conditions mentioned in the previous sub-section. Let us explain one condition of through lane vehicles whose statuses change from *stopped* to *crossing* based on the following rewrite rule:

```
rl [enter-fairN-T] :
 {(v[v(I)]: through,stopped) (lane[through]: (v(I) ; TQ))
 (lane[nonThrough]: empq) (mode: fair)
 (turn: nonThrough) (crossing?: false) OCs}
 => {(v[v(I)]: through,crossing) (mode: fair)
 (lane[nonThrough]: empq) (turn: nonThrough)
 (lane[through]: (v(I) ; TQ)) (crossing?: true) OCs} .
```

where `v(I)` and `TQ` are Maude variables for real vehicles and queues, respectively. The rewrite rule says, if `mode` is `fair`, `turn` is `nonThrough`, `crossing?` is `false`, no vehicles in the non-through lane, and the top queue is `v(I)` whose status is `stopped`, then `v(I)` changes its status to `crossing`, which means `v(I)` is in the merge point, and update `crossing?` to `true`. The other rewrite rules can be specified likewise, please refer to [10] in detail.

III. Designing a state picture template of the AR Protocol

A. Idea

As written, there are many possible ways to design a state picture template. However, if the design of state picture template is simple, it is hard to conjecture characteristics when observing animations produced by such state picture template [4], [13]. Therefore, our purpose mainly focuses on explaining how to design the state picture template. Then, we show some factors affecting the design of the state picture template in the next sub-section.

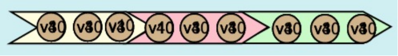
Firstly, some work [4], [5], [13] have pointed out the usefulness of their state picture templates and also given some tips to make a good state picture template. In the present paper, we mainly use some of such tips for our design and summarize them as follows:

- Values of observable components should be visualized as much as possible.
- When an observable component has only two kinds of values, it should be visually/graphically represented as a light bulb.
- If one observable component has a value that contains more than one component inside, this value is called a composite value. For example,

(`v[v[`i`]]`: *laneInfo*, *veStat*) is an observable component that contains a composite data (*laneInfo*, *veStat* are two components). We should carefully select observable components and component values of composite data to visualize.

- If a value of an observable component does not change, it should be expressed as a fixed label.

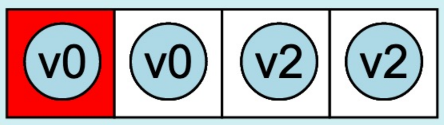
Based on the tips, we first carefully select some observable components to visualize. `#ucvs` and `crossing?` are not concerned to visualize because we can get them via observing other observable components, such as observing the merge point when some vehicle is located, we can obtain the value of `crossing?`. `turn`, `gstat`, and `mode` are considered to visualize as a light bulb because they contain two kinds of values. *laneInfo* of vehicles (such as *through* and *nonThrough*) is visualized as a fixed label. The remaining observable components are two queues and vehicle statuses that are main parts of the protocol. One possible way to visualize the queues is to use the visualization of the work [4] as follows:



where brown circles represent vehicles, numbers in the circles represent ids of vehicles, colors of each area in the arrow present statuses of vehicles such as *approaching*, *stopped*, and *crossing* from left to right, respectively. There is one property (or assumption) of the protocol in [5] that is mainly different from the AR protocol as follows:

> In the same lane, some vehicles can have the same status value. For example, two vehicles may have the same status *crossing* at the same time. However it is prohibited in the AR protocol.

Therefore, they [5] need to prepare each area that can contain enough positions for the maximum number of vehicles participating in the lane, such as three vehicles for each area in the above figure. In the present paper, we propose the design of queue visualization to optimize such positions using a similar way to the visualization technique of [5] and analogous display (an existed technique of SMGA [1]). The following figure describes our design of the queue:

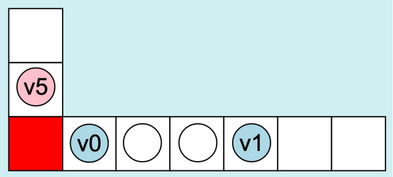


where circles present vehicles; each square displays one element in the queue. If no element is in the queue, no circle is displayed. From the left to right, the first square whose color is red presents the merge point that contains vehicles whose statuses are *crossing*. The remaining squares contain vehicles whose statuses are *stopped* and *approaching*, and spaces whose statuses are *space* (for the through lane). Note, in each square, there are at least three circles representing three vehicles (not included their values). When some node is in the merge point, we use analogous display for this case. We use the visualization technique of [5] for other cases in which there is no node in the merge point, please refer [5] in detail.

Furthermore, by observing graphical animations based on some drafts, we comprehend that when the mode is prioritized, the observable component `turn` does not affect vehicles entering the merge point. Therefore, we design two observable components `turn` and `mode` following to this attribute. The idea is that when the mode is prioritized, we do not let the observable component `turn` display. In other words, we use a trick to display some symbols and hide the observable component `turn` when the mode is prioritized. We explain this idea in detail in the next sub-section. Note, to design the state picture template in the present paper, we use some laws of the Gestalt principles [11], [12] (such as, the similarity law - using same shapes, colors or sizes to same values). It is one factor that affects the state picture template and we will show its usefulness in the next section. How to design all observable components are mentioned in the next sub-section.

B. Designing a State Picture Template

Fig. 2 shows fully our state picture template. We suppose that there are four vehicles and two spaces participating in the through lane, and two vehicles participating in the non-through lane. The main part of the state picture template is two lanes represented by vertical and horizontal rectangles that are formed by some squares. The vertical one represents the non-through lane while the horizontal one represents the through lane. The red square that is the intersection of two lanes refers to the merge point. Circles with numbers inside correspond to vehicles while blank circles correspond to spaces in the through lane. Light-blue and light-yellow circles are vehicles in the through and the non-through lane, respectively. Light-pink circles are vehicles whose statuses are *stopped*. The following figure shows two lanes when the value of the observable component `lane[through]` is `v(0);dv(0);dv(1);v(1)` and the value of the observable component `lane[nonThrough]` is `v(5)`.



where the status of `v(5)` is *stopped*; two vehicles `v(0)` and `v(1)` whose statuses are *approaching*; two spaces `dv(0)` and `dv(1)` whose statuses are *space*.

There are some arrows that close to two lanes representing two main parts: (i) the directions and (ii) the turns of two

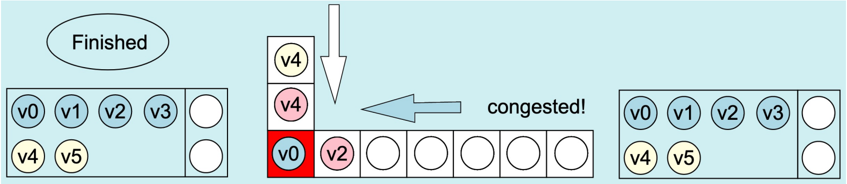


Figure 2. A state picture template for the AR protocol (1)

lanes. Intuitively, the shapes of the arrows refer to (i) while the colors of the arrows refer to (ii). Note, the arrow that nears to the through lane includes light-blue and white arrows while the arrow that nears to the non-through lane consists of light-yellow and white arrows. The light-blue and the light-yellow arrows indicate the turn of through lane and non-through lane vehicles, respectively. The white arrows are used as a trick to hide the turn. In the prioritized mode, the light-blue arrow (nearing the through lane) and the white arrow (nearing the non-through lane) are displayed. In the fair mode, the text "congested!" is displayed and the arrows are displayed following the observable component `turn`. All arrows and the text are visualized using the analogous display.

The rectangle in the left-side of Fig. 2 contains the vehicles and the spaces whose statues are $crossed$ and $yield$, respectively. The rectangle in the right-side of Fig. 2 contains the vehicles and the spaces whose statues are $running$ and $unspace$, respectively. The ellipse with the text "Finished" inside refers to the observable component `gStat`. When the value of `gStat` is *fin*, the ellipse is displayed; otherwise, it is not displayed. Observable components in this paragraph are visualized using the analogous display.

Finally, to extend the state picture template for more cases (such as there are five vehicles and four spaces in the through lane), it is not difficult to redesign the state picture template with our proposal. Users can prepare more squares such that the number of squares reaches the worst case that may occur. For example, there are five vehicles and five spaces participating in the through lane, we need to prepare 11 positions (including the merge point) in case all of them are put into the queue. We design the state picture template shown in Fig. 3 for a case in which there are five vehicles and five spaces in the through lane and five vehicles in the non-through lane. Users can utilize it in case the number of vehicles is up to five. Note that we fix the id of vehicles in the through and the non-through lane from `0` to `4` and `5` to `9`, respectively. Therefore, users need to config the initial state for such a restriction. Note also that five (vehicles and spaces) is an enough number (not wait so long) to use Maude to confirm characteristics [10].

IV. Confirmation of Guessed Characteristics of the AR Protocol and Some Lessons Learned

A. Guessing Some Characteristics

Let us repeat that observing graphical animations based on some drafts helps us comprehend one attribute of the AR protocol. Observing graphical animations based on our proposal also helps us understand another attribute shown in Fig. 4. Note, in Fig. 4, two state pictures in the top and two state pictures in the bottom of the figure are from two different state sequences. Two attributes are described as follows:

- Characteristic 0: The turn is not concerned when the protocol is in the prioritized mode.
- Characteristic 0': There exists a case such that there is one vehicle whose status is $approaching$ in the non-through lane, but this status changes to $stopped$ even no vehicle is in the through lane.

Both characteristics are the intention of the AR protocol, however, observing graphical animations helps us better comprehend the protocol by obtaining these attributes.

To conjecture some other characteristics, we use some tips [4]. These tips say that focusing on one or two observable components can help us guess its relations. We focus on two lanes and conjecture some characteristics as follows:

- Characteristic 1: There is at most one vehicle whose status is $stopped$ in each lane.
- Characteristic 2: When two vehicles whose statuses are $stopped$ in both lanes, no vehicle is in the merge point.
- Characteristic 3: There are at most two vehicles whose statuses are $stopped$ in the protocol.

To conjecture the characteristics above, color is a main factor. Based on the colors designed by the similariry law of the Gestalt principle, we can observe that at most one light-pink color in each lane shown in Fig. 5. The characteristic 3 can be conjectured by two characteristics 1 and 2. The following characteristics are conjectured by focusing on a vehicle whose status is $stopped$.

- Characteristic 4.1: There is one vehicle whose status is $stopped$ in the non-through lane and if some vehicle is in the merge point, then this vehicle is in the through lane or the through lane is not empty.
- Characteristic 4.2: There is one vehicle whose status is $stopped$ in the through lane and if some vehicle is in

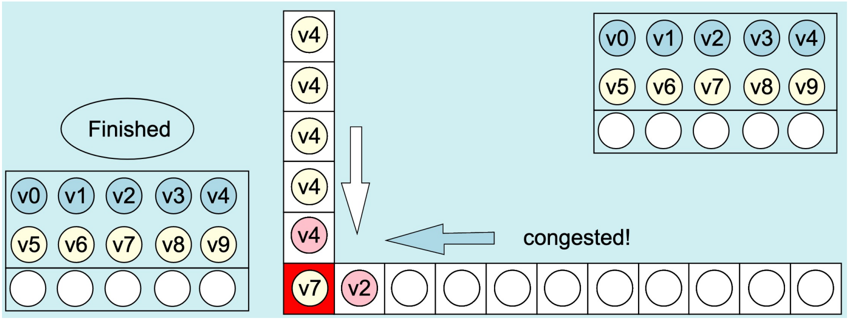


Figure 3. A state picture template for the AR protocol (2)

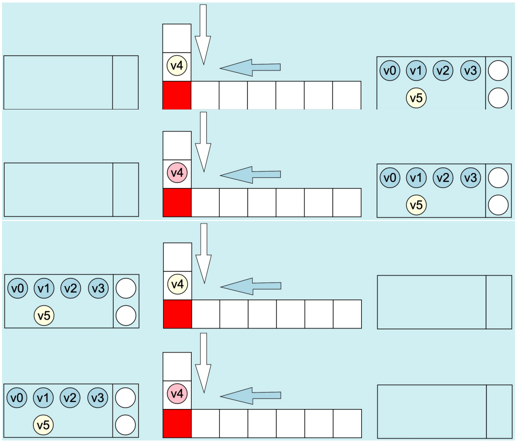


Figure 4. Some state pictures (1)

the merge point, then this vehicle is in the non-through lane or the non-through lane is not empty.

B. Confirmation of Guessed Characteristics

By using the search command of Maude, we can confirm the characteristic 1 by the command as follows:

```
search [1] in AVMP : init =>*
{(v[X:Vid]: l1:Lane, stopped)
(v[Y:Vid]: l1:Lane, stopped) OCs:Soup{OComp}} .
```

where `AVMP` is the name of Maude module; `X:Vid` and `Y` are vehicle IDs; `l1:Lane` is sort of lane; and `OCs:Soup{OComp}` refers to other observable components. The search command tries to find a reachable state that satisfies the pattern. It does not return any solution so that the characteristic is confirmed.

To confirm the characteristic 2 and 3, we use two commands as follows:

```
search [1] in AVMP :
init =>* {(v[X:Vid]: through, stopped) (crossing?: true)
(v[Y:Vid]: nonThrough, stopped) OCs:Soup{OComp}} .

search [1] in AVMP :
init =>* {(v[X:Vid]: L1:Lane, stopped)
(v[Y:Vid]: L2:Lane, stopped)
(v[Z:Vid]: L3:Lane, stopped)
OCs:Soup{OComp}} .
```

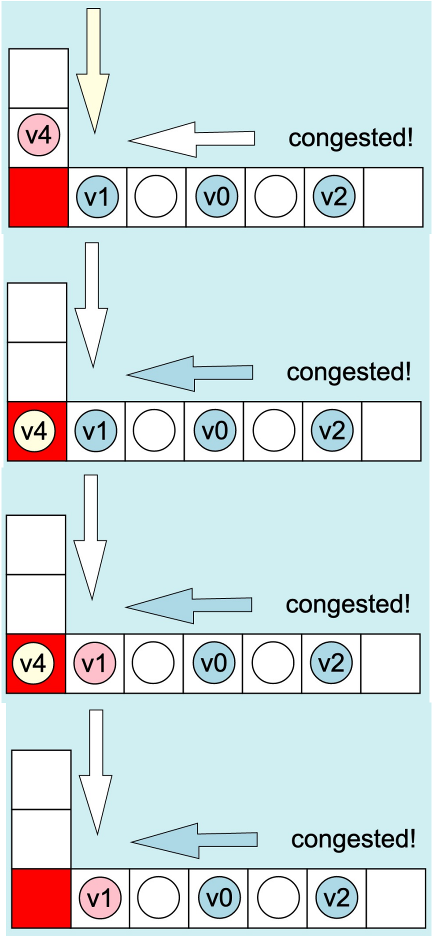


Figure 5. Some state pictures (2)

Each search command tries to find a reachable state that satisfies the pattern. They do not return any solution so that two characteristics are confirmed.

To confirm the characteristics 4.1 and 4.2, we use the following commands:

```
search [1] in AVMP :
init =>* {(v[X:Vid]: L:Lane, crossing)
(lane[through]: Q:Queue{Vid})
(crossing?: true)
(v[Y:Vid]: nonThrough, stopped) OCs:Soup{OComp}}
such that L:Lane =/= through or Q:Queue{Vid} == empq .

search [1] in AVMP :
init =>* {(v[X:Vid]: through, stopped) (crossing?: true)
(v[Y:Vid]: nonThrough, stopped) OCs:Soup{OComp}} .
```

Those commands above also do not return any solution so that the characteristics 4.1 and 4.2 are confirmed.

Note that we need to confirm all guessed characteristics even though we intuitively know that they are correct. One flawed characteristic we have conjectured is that there is a non-through lane vehicle whose status is $approaching$, the status always changes to $stopped$ after that. However, when we use Maude LTL model checking to confirm it, the model checker returns a counterexample. The counterexample says that the status of non-through lane vehicles can change to $crossing$ from $approaching$ when the mode is $fair$ and the turn is $nonThrough$.

C. Lessons Learned

As discussed, designing state picture templates is not a straightforward task for SMGA. When starting to design a state picture template, some tips from some previous work can be used as guidelines but the use of tips still mainly depends on the properties or assumptions of protocols. Therefore, we need to carefully select some existing tips as guidelines to make some first drafts. Then, we redesign some drafts based on properties via observing graphical animations or understanding the protocols. Note that the better we understand the protocol, the more we can design effectively the state picture template. To this end, we summarize our experiences as a procedure on how to design the state picture template. The procedure is as follows:

1. Consulting or selecting some previous tips as guidelines to design some first drafts of the state picture template.
2. Using some laws of the Gestalt principle, especially the similarity law to design values of observable components, such as we color statuses of vehicles in both lanes.
3. Combining some existing visualization techniques to design some observable components in case one visualization technique is not enough. For example, we use the analogous display and queue visualization technique of [5] to design queue data structure.
4. If possible, observing graphical animations of some draft versions to understand some properties or assumptions of the protocols and then updating the drafts.

All state picture templates and inputs used in the paper are available at the website [1].

V. Conclusion

We have graphically animated the AR protocol by using SMGA in which we describe in detail how to make a state picture template from scratch. To design the state picture templates, we have combined kinds of visualization supported by SMGA, and used the tip [4] and Gestalt principles [11], [12]. Observing graphical animations help us comprehend some properties or assumptions of the AR protocol and guess some characteristics of the protocol. Those characteristics are also confirmed with the searfch command of Maude. We have also summarized our experiences as a practical procedure that can be used to make a state picture template from scratch. One piece of our future work is to prove guessed characteristics by writing proof scores in CafeOBJ [14].

References

[1] T. T. T. Nguyen and K. Ogata, "Graphical animations of state machines," in *15th DASC*, 2017, pp. 604–611.

[2] M. T. Aung, et al., "Guessing, model checking and theorem proving of state machine properties – a case study on Qlock," *IJSECS*, vol. 4, no. 2, pp. 1–18, 2018.

[3] T. W. Mon, et al., "Graphical animations of the ns(l)pk authentication protocols," *JVLC*, vol. 2021, no. 2, pp. 39–51, 2021.

[4] D. D. Bui and K. Ogata, "Better state pictures facilitating state machine characteristic conjecture," *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 237–272, 2022.

[5] D. D. Bui, et al., "Graphical animations of the Lim-Jeong-Park-Lee autonomous vehicle intersection control protocol," *JVLC*, vol. 2022, no. 1, pp. 1–15, 2022.

[6] Frank v. H., et al., "Interactive visualization of state transition systems," *IEEE Transaction on Visual and Computer Graphics*, vol. 8, no. 4, pp. 319–329, 2002.

[7] J. Lim, et al., "An efficient distributed mutual exclusion algorithm for intersection traffic control," *J. Supercomput.*, vol. 74, no. 3, pp. 1090–1107, 2018.

[8] S. Aoki and R. R. Rajkumar, "A merging protocol for self-driving vehicles," in *ICCPS*, 2017, p. 219–228.

[9] Liu. M, et al., "Formal specification and model checking of an autonomous vehicle merging protocol," in *QRS-C*, 2021, pp. 333–342.

[10] Ware and Colin, *Information Visualization: Perception for Design*, 3rd ed. Morgan Kaufmann, 2012.

[11] D. Todorovic, "Gestalt principles," *Scholarpedia*, vol. 3, no. 12, p. 5345, 2008.

[12] D. D. Bui and K. Ogata, "Graphical animations of the Suzuki-Kasami distributed mutual exclusion protocol," *JVLC*, vol. 2019, no. 2, pp. 105–115, 2019.

[13] R. Diaconescu and K. Futatsugi, *CafeOBJ Report*. World Scientific, 1998.

[1] https://bddang.bitbucket.io

Brain tumors classification from MRI images:
A comparative study between different neural networks

Bernardo Breve, Loredana Caruccio, Gaetano Cimino, Stefano Cirillo,
Gianpaolo Iuliano, Giuseppe Polese

Department of Computer Science
University of Salerno, Italy
{bbreve,lcaruccio,gcimino,scirillo,gpolese}@unisa.it
{g.iuliano21}@studenti.unisa.it

Abstract

The detection of brain tumors through the analysis of images is becoming increasingly common for promptly treating patients. Among the different types of imaging techniques, Magnetic Resonances Imaging (MRI) is probably the most popular one in the pre- and post-treatment to estimate the structure of tumors. Thus, they also represent a useful means for supporting intelligent techniques in the identification of brain tumors, enabling machine learning models to completely automate the classification task. In this paper, we propose a new methodology for classifying brain tumors through the analysis of MRI images. In particular, our approach relies on a feature extraction technique to obtain representative data, which are used as input for two predictive models, a Convolutional Neural Network (CNN) and a Residual Neural Network (ResNet). We discuss experimental evaluation performed over a ground-truth dataset and show a comparative analysis between proposed models in the classification of tumors according to their type.

Index terms— Magnetic Resonance Imaging, Brain Tumor, Convolutional Neural Networks, AI Healthcare.

1 Introduction

The brain is the largest and the most complex organ of the human body, where more than ten billion brain cells work synergistically. Unfortunately, some of these cells could form an abnormal group, compromising the usual brain functionalities and also damaging the remaining cells. Such a group of cells (or nodule) represents a tumor.

According to a study published in 2017, the overall incidence rate of malignant brain tumors is somewhat worrying [16]. In fact, statistical data show that the incidence rate of brain tumors in Central Europe is 2.37% for children aging from 0 to 14 years, 2.73% in the range 15-39 years old, and 15.06% for adults aging over 40 years, on a sample of $100,000$ people. This data highlights the necessity to support clinicians in the identification of such disease, and in its characterization. In particular, the automatic identification of the tumor type not only permits to assist radiologists, but also to have insights before applying histological biopsies [9]. To this end, the Magnetic Resonance Imaging (MRI) appears to be one of the most efficient imaging techniques for the detection of brain tumors. Thanks to the high resolution provided by these images, they can provide a lot of information on both the structure of the brain and the presence of abnormalities in the brain tissues. By exploiting the potential of the output images provided by MRI, many machine learning models for the classification of brain tumors have been proposed in the literature. Techniques used in classification models include Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), and Gray Level Co-Occurrence Matrix (GLCM) for the feature extraction [7]. Techniques for the segmentation of tumor masses in MRI images include Fuzzy C-Means (FCM), K-Means, DBSCAN, U-Net, and image manipulation techniques [30, 10]. Finally, techniques for the classification include Support Vector Machine (SVM), Decision Tree, and MultiLayer Perceptron (MLP). However, in the last decade, Convolutional Neural Networks (CNNs) and Residual Neural Networks (ResNets) obtained considerable consensus for the image classification tasks [12].

In this paper, we propose a new methodology for the classification of brain tumors, by also performing a com-

DOI reference number: 10.18293/DMSVIVA22-010

parative evaluation on two of the models for image classification, i.e., CNN and ResNet, evaluating their performances in terms of Precision, Recall, and F1-Score. In particular, traditional CNNs represent the most employed kind of network for the brain tumor classification task. Instead, although ResNets have been widely applied for image classification, they are quite novel for this task [29]. Thus, we defined both a CNN and a ResNet model, and trained them on an open-source dataset of MRI images labeled according to three classes of tumor types. Moreover, images have been passed through a pre-processing phase aimed at highlighting the tumor cells with respect to other tissues represented within an MRI image. This process is particularly important since classification algorithms cannot work well if the tumor is not accurately extracted and isolated.

The paper is organized as follows. In Section 2 we present related works proposed in the literature. Then, in Section 3 we provide details of the considered dataset and the MRI images on which the classification task is performed. In Section 4 we present the architectures of the defined models, whereas in Section 5 we describe the classification methodology. Finally, conclusions and future research directions are discussed in Section 6.

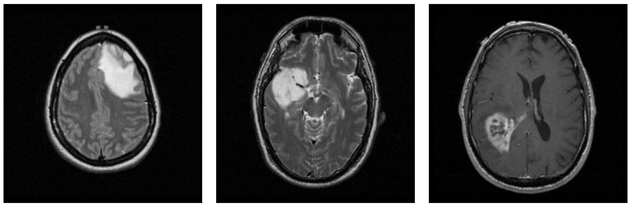
2 Related Work

The realization of powerful machine learning models for classifying brain tumors is receiving particular attention in the scientific community. Thus, several techniques and methodologies have been proposed in the literature to improve the state of the art of current classifiers. *Sanjeec Kumarl et al.* have proposed a hybrid approach for the classification of brain tumors starting from MRI images [15]. The methodology involves the Discrete Wavelet Transformation (DWT) as a feature extraction technique, genetic algorithms to reduce the number of features, and the Support Vector Machine (SVM) model for classifying the type of brain tumor. SVM model has also been used in [27].

Rajeshwar Nalbalwar et al. proposed a methodology that involves several pre-processing techniques, such as equalization of the histogram, image segmentation, and the GLCM to collect features that are employed for training an Artificial Neural Network (ANN) [21]. Similar pre-processing phases are used in conjunction with the Neuro-Fuzzy [13], Naive Bayes, and Decision Tree classifiers [20].

In general, many types of ANN models have been applied for the classification of brain tumors, such as Probabilistic Neural Networks [8, 23], Pulse-Coupled Neural Networks [14], and Back-Propagation Neural Networks [28, 24]. Instead, a Deep Learning-based method has been proposed in [18], where the DWT has been used as a feature extraction technique.

Recently, the CNN model represents the most popular technique for classifying brain tumors from MRI images [2, 22]. In fact, the convolutional layers involved in CNN models provide an effective feature extraction process. The CNN designed by *Chang et al.* predicts the molecular genetic mutation status in gliomas using the PCA for the extraction of the classification features [6]. In the survey conducted at Stanford University [26], a CNN has been applied for the classification of the glioma types from MRI images, which uses a pre-trained Neural Network model dealing with the sub-division of tumor/non-tumor parts.



(a) Oligodendroglioma (b) Astrocytoma (c) Glioblastoma

Figure 1: The Analyzed Tumors.

In this paper, we propose a process to improve the segmentation of tumors within images in order to infer significant features for training a CNN and a ResNet model. Thus, this work is aimed not only at the definition of high-performing models, but also at a comparative evaluation that can reveal in which cases the models outperform each other.

3 Processing Tumor Images

In order to evaluate the proposed solution, we have employed the Repository for Molecular Brain Neoplasia Data (REMBRANDT) of the Cancer Imaging Archive [11]. Currently, it is the only open-source dataset with a large number of images equipped with patient metadata. The REMBRANDT dataset has been developed by the National Cancer Institute (NCI) at the request of the National Institutes of Healths and the National Institute of Neurological Disorders and Stroke. In particular, it consists of images in DICOM format (a standard format for MRI images) labeled with four different classes: Oligodendroglioma (Figure 1a), Astrocytoma (Figure 1b), Glioblastoma (Figure 1c), and unidentified tumors.

The dataset labels refer to types of tumors identified from the clinical records of 130 patients. The data collection projects the anatomy of the brain in i) transversal (or commonly called axial or horizontal), ii) sagittal (also known as anteroposterior), and iii) coronal (also known as frontal or lateral). Moreover, it considers different types of images among the most used MRI sequences, such as T1-Weighted, T2-Weighted, FLAIR, and PERFUSION.

Projection type	Axial	
Sequence type	T1-Weighted T2-Weighted FLAIR PERFUSION	
Metadata	yes	
Classes	**Training images**	**Test images**
Astrocytoma	1400	100
Glioblastoma	1400	100
Oligodendroglioma	1400	100

Table 1: An overview of the structure of the dataset.

In this work, we do not consider images of unidentified tumors and those for which we have encountered the presence of noise. Moreover, to avoid the overfitting problem of the proposed models, we have applied an undersampling approach by randomly selecting a fair number of samples for each class of brain tumors. Thus, we have considered 1500 MRI images for each category, resulting in a dataset of 4500 images. Finally, we have divided the dataset into training and test sets. In particular, the test set is composed of 300 randomly sampled images, i.e., 100 images for each type of tumor. Details of the dataset and experimental setup are shown in Table 1.

4 The Proposed Neural Networks

Neural Networks represent one of the most used predictive approaches in computer vision for processing images and multimedia data. It has been proved that they represent an extremely useful tool in different scenarios, ranging from the autonomous driving of cars and drones [19, 25] and financial forecasting [3], to the detection of medical diagnoses for supporting and promptly treating patients [5, 17]. The architecture of neural networks for a classification task is composed of one or multiple layers of neurons that permits to efficiently obtain final predictions. In this paper, we adopt two different neural networks to identify brain tumors from MRI images, a Convolutional Neural Network and a Residual Neural Network, respectively.

Convolutional Neural Network (CNN). The architecture of the CNN that we have designed for the classification of cerebral tumors is shown in Figure 2.

Images are resized to 32×32 pixels before being sent in input to the first convolutional layer. The latter uses 12 filters to extract features from images. In particular, in order to search the entire matrix, the applied filters employ a kernel of size 5×5. Moreover, it uses the Leaky ReLU activation function with a parameter set to 0.1.

The output of the first convolution layer is represented by a tensor of size $28 \times 28 \times 12$. This is sent to a Max Pooling layer, through which it is possible to reduce the matrix dimensions, and obtain summarized values for sub-regions of dimension 4×4.

The output of the Max Pooling function is supplied as input to the second convolutional layer having 12 filters, a kernel of size 3×3, and again a Leaky Relu function. The result is a tensor of size $5 \times 5 \times 12$. The Max Pooling function associated with the second convolutional layer has a 2×2 kernel size, providing in output a tensor of size $3 \times 3 \times 12$. The output is further reduced by using the Global MaxPooling function with a tensor of size 12.

At this point, the first Fully connected layer is activated (also called Dense layer), which is a hidden layer where each node is completely connected only to the nodes contained in the next hidden layer. In the first Fully connected

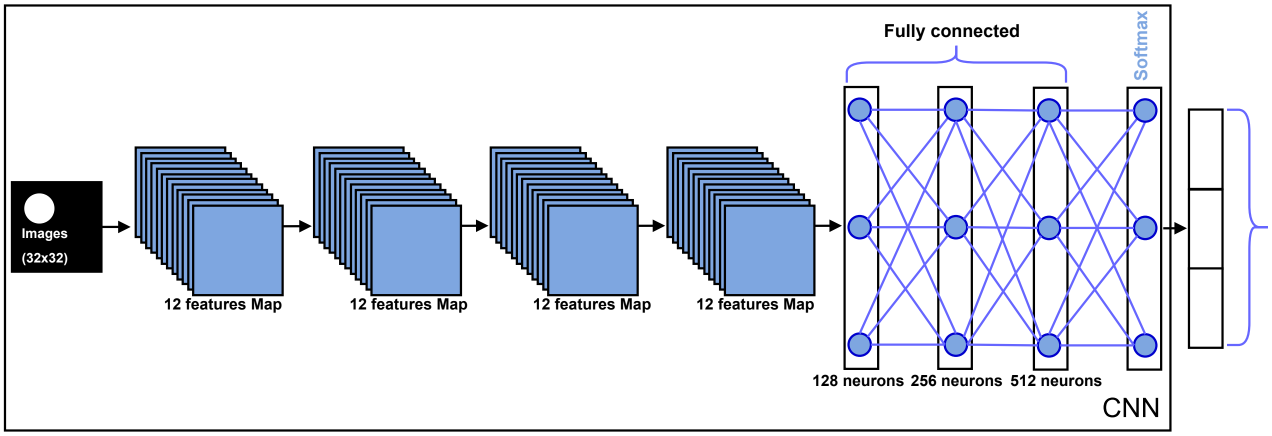


Figure 2: Architecture of the proposed Convolutional Neural Network (CNN).

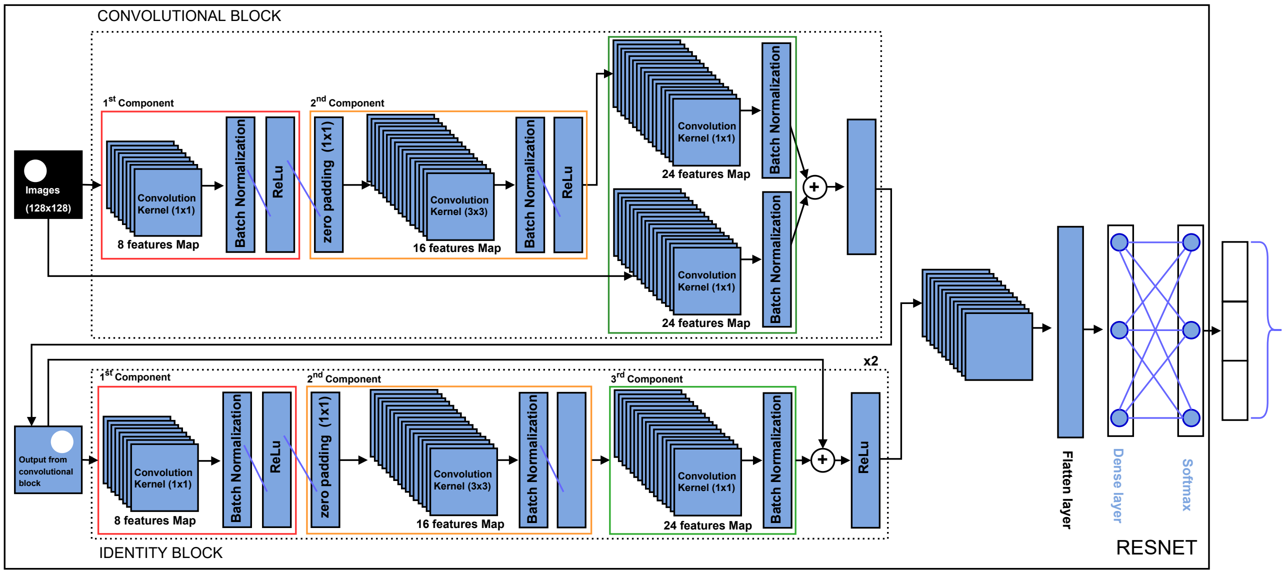


Figure 3: Architecture of the proposed Residual Neural Network (ResNet).

Layer, the model uses 128 neurons, the LeakyRelu activation function, and a DropOut function with a rate of 0.5. The second Fully connected Layer analyses the results of the previous layer by means of 256 assigned neurons, the LeakyRelu activation function, and the DropOut function with a rate of 0.25. Finally, the output is sent to a third Fully connected Layer with 512 neurons, a LeakyRelu activation function, and a DropOut function with a rate of 0.25. In order to categorize each result in one of the predefined classes, the network uses the Softmax activation function. Moreover, the Adam optimizer has been adopted, by setting a learning rate to 0.0001. Finally, to calculate the degree of error in the training between the computed outputs and the expected ones, the categorical-cross-entropy loss function has been used, since it allows to calculate the error in a multi-class classification task.

Residual Neural Network (ResNet). The second neural network designed for our study is a ResNet.

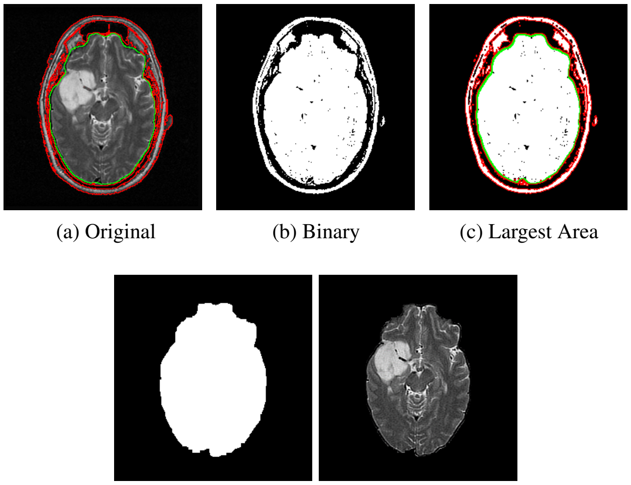
The architecture of the ResNet is shown in Figure 3. It is composed of a Convolutional block, two Identity blocks, an AveragePooling layer, and several Fully connected layers. The Convolutional block is the first block of the ResNet, which receives images with dimensions 128×128 as input.

As shown in Figure 3, the convolutional block receives an image as input and processes it by means of three different components. The first component includes a Convolutional layer that uses a kernel 1×1 and 8 feature maps, a batch normalization layer, and the ReLu activation function. The output of the first component is read from the second component that includes a Zero Padding layer, with filter 1×1, a convolutional layer that uses a kernel 3×3 and 16 feature Maps, a batch normalization layer, and the ReLu activation function. Instead, the third component is divided into two parts. Both parts include a Convolutional layer that uses a kernel 1×1 and 24 features Maps, and a Batch normalization layer. The first part processes the output of the second component, while the second part works with original input images. Finally, the Convolutional block uses ReLu activation function for returning the processed image.

After the Convolutional block, the proposed ResNet is composed of two Identity blocks, which have a similar structure to the Convolutional blocks, except for the third component. In fact, this component includes a Convolutional layer that uses kernel 1×1 and 24 features Maps, and a Batch normalization layer. Finally, each Identity block uses the ReLu activation function for returning the processed image.

The output of the Identity block is read from an Average Pooling layer that uses a filter 3×3 for processing the image. After the pooling step, a Flatten layer prepares the data for the Fully connected layer, which will then produce three outputs evaluated by the SoftMax activation function.

For this network, the learning rate has been set to 0.001 with Adam optimizer, while the *clipnorm* has been set to 1. Similar to the CNN designed for brain tumor detection, the ResNet uses categorical cross-entropy as a loss function. Moreover, in order to reduce the overfitting problem, it was used the L2 regularizer (Ridge Regression) with the parameter λ set to 0.005.



(a) Original (b) Binary (c) Largest Area

(d) Mask (e) Result

Figure 4: Skull Stripping.

5 The Complete Classification Process

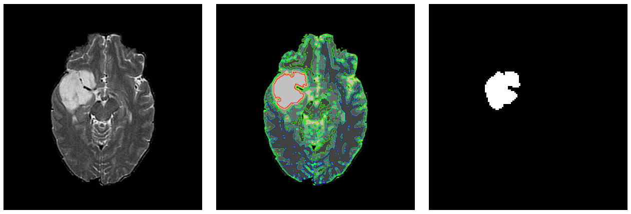
In this Section, we describe the process performed for classifying brain tumors. In particular, we first describe image pre-processing steps performed for identifying the sections of the images containing brain tumors (Section 5.1). Then, we discuss the evaluation step of the CNN and ResNet, and analyze their experimental results (Section 5.3 and 5.4).

5.1 Brain Tumor Extraction

The first step has concerned the isolation of the brain by removing non-cerebral tissues (i.e., the skull stripping), and the detection of the tumor (i.e., the tumor segmentation). In general, these operations play a fundamental role in the classification process, since a good segmentation contributes to achieving better classification results.

Skull Stripping. This operation allows the removal of non-cerebral tissues, such as the skull and the scalp, from the MRI images (i.e., the red contours in Figure 4a). In fact, their presence can negatively affect tumor detection, since non-cerebral tissues usually have high-intensity colors.

The image is first converted from a gray to a binary representation (Figure 4b). Then, the two largest blobs, which refer to the skull and the brain, are extracted from the image, so as to isolate them from small blobs, which represent noise. Successively, the image is expanded, aiming to stretch the objects within the image. Thus, the largest area (Figure 4c) is extracted from the image (it corresponds to the brain), becoming the used cerebral mask (Figure 4d). Finally, the extraction of the brain (Figure 4e) is accomplished by overlapping the mask with the original image.



(a) Pre Fuzzy C-Means (b) Post Fuzzy C-Means (c) Binarization

Figure 5: Tumor Segmentation.

Tumor Segmentation. The extraction of the tumor from the brain is one of the most important operations. In order to make a suitable segmentation of the tumor, in our classification process, we applied the *Fuzzy C-Means* clustering algorithm, by using 5 clusters, and deriving the tumor area (outlined with a red border in Figure 5b) by considering the pixels with the highest degree of intensity. The number of the optimal clusters to adopt for the *Fuzzy C-Means* was derived by using the *Elbow method* [4] and the results are shown in Figure 6. The isolation of the tumor was accomplished by converting the image into a binary format, by assigning 1 (white color) to the pixels belonging to the cluster with the highest intensity, and 0 (black color) to the remaining ones.

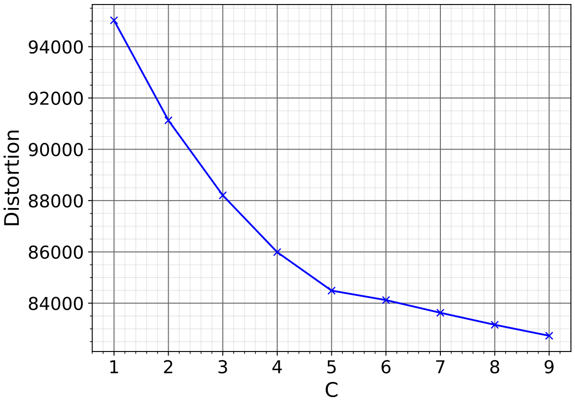


Figure 6: The Elbow method showing the optimal C.

5.2 Evaluation Metrics

The performances of CNN and ResNet have been evaluated in terms of Precision, Recall, Accuracy, and F1-Measure. In particular, let TP, TN, FP, and FN be the values of True Positive, True Negative, False Positive, and False

Negative, respectively, then we can define the evaluation metrics as follow:

- **Precision** represents the number of positives that are correct, over all identified positives: $\text{Prec} = \frac{\text{TP}}{\text{TP+FP}}$
- **Recall** represents the proportion of positives that are correctly identified: $\text{Rec} = \frac{\text{TP}}{\text{TP+FN}}$
- **Accuracy** represents the number of positives and negatives correctly identified over the total number of tested elements: $\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$
- **F1-score** represents the armonic mean of precision and recall: $\text{F1} = 2 \times \frac{\text{Prec} \times \text{Rec}}{\text{Prec+Rec}}$

5.3 Evaluating CNN Performances

The first technique adopted for validating the CNN architecture described above is the K-Fold Cross Validation [1], which permits the estimation of the effectiveness of a predictive model by also highlighting possible overfitting. In particular, we have divided the dataset into 6 folds with 700 images, each considered one time as a validation set. Table 2 shows the classification accuracy achieved in each iteration, the standard deviation, and the average of the values. It is possible to notice that in all the folds, we have achieved accuracy values higher than 92%, with a maximum value equal to 95.04%.

After cross-validation, we have analyzed the learning and the information loss curves of the proposed CNN over the entire training set. The learning curve showed no irregularities, hence no overfitting. Similar results have been achieved for the loss of information curve, which follows a decreasing trend. During the evaluation, the CNN achieved an accuracy of 96%.

	Accuracy (%)
Fold 1	95.04
Fold 2	92.98
Fold 3	93.11
Fold 4	92.84
Fold 5	94.05
Fold 6	93.22
Standard Deviation	0.77
Average	93.54

Table 2: Overview of the results achieved from the K-Fold Cross Validation.

Figure 7 shows the confusion matrix of CNN. As we can see, the CNN reached maximum score values for the identification of *Astrocytoma* and *Glioblastoma* tumors, whereas for *Oligodendroglioma*, it wrongly classifies 12 of the 100 images involved in the evaluation.

5.4 Evaluating ResNet Performances

The ResNet involved in our study has been trained considering 1120 images for each type of tumor, 100 images for the testing, and 280 images for the validation step. Moreover, this neural network has been trained for 100 epochs, with a batch size equal to 64.

The validation phase has provided an average accuracy score of 95% with a peak of 97%, across the epochs.

Figure 8 shows the confusion matrix of the proposed ResNet. As we can see, ResNet is able to correctly classify most images, especially those representing the *Astrocytoma* tumor. However, although the results for *Glioblastoma* and *Oligodendroglioma* are still high, ResNet has more difficulty in identifying these types of tumors.

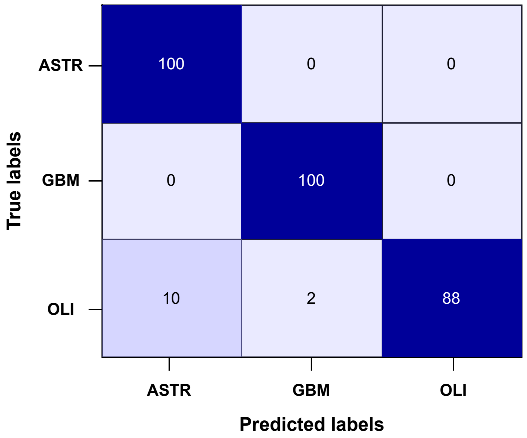


Figure 7: Confusion Matrix of CNN.

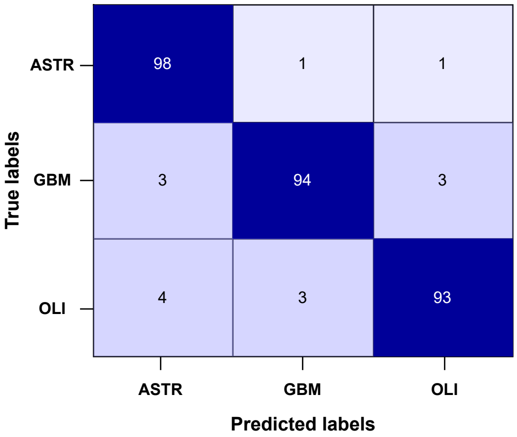


Figure 8: Confusion Matrix of ResNet.

	Precision (%)		
Models	ASTR	GBM	OLI
CNN	0.91	0.98	1.00
ResNet	0.93	0.96	0.96

Table 3: Precision of the classification models.

	Recall (%)		
Models	ASTR	GBM	OLI
CNN	1.00	1.00	0.88
ResNet	0.98	0.94	0.93

Table 4: Recall of the classification models.

	F1 - Score (%)		
Models	ASTR	GBM	OLI
CNN	0.95	0.99	0.94
ResNet	0.96	0.95	0.94

Table 5: F1-Score of the classification models.

5.5 Comparative Evaluation

Performances obtained by the designed neural networks have been compared by computing the accuracy, precision, recall, and F1-score metrics (see Section 5.2). In particular, after the evaluation step, the accuracy achieved by the CNN is equal to 96%, whereas 95% by the ResNet.

Tables 3 and 4 show the precision and recall values for both neural networks. As we can see from Table 4, the CNN correctly classifies all images of *Astrocytoma* and *Glioblastoma* tumors, while with *Oligodendroglioma*, it achieves a recall of 88% compared to 93% obtained with the ResNet. In fact, CNN outputs more frequently the class *Astrocytoma* in case of uncertainty with respect to the *Oligodendroglioma* tumors (see Figure 7). Consequently, the precision value for the *Astrocytoma* class decreases, as highlighted in Table 3. Thus, the ResNet outperforms the CNN in terms of precision with *Astrocytoma* tumor, while with the other types of tumors, the CNN outperforms ResNet.

Finally, the results of F1-score metrics are shown in Table 5. As we can see, the ResNet outperforms the CNN with *Astrocytoma* tumor, and achieves similar results to CNN with *Oligodendroglioma* tumor.

6 Conclusion

MRI-based brain tumor classification is an important and interesting problem considering its relevance for health care. To this end, in this paper, we have proposed a classification methodology relying on an automated image preprocessing step. Moreover, we proposed two artificial neural networks, a traditional CNN and a ResNet, which have been then considered to perform a comparative evaluation. Results demonstrated that both models achieved high performances, even though with different results. In fact, the CNN model performed particularly well on the classification of *Astrocystoma* and *Glioblastoma* classes. On the other hand, although the ResNet model accuracy is slightly lower than the CNN one, the model resulted more consistent over all the classes, as proved by the F1-score values.

In the future, we would like to test the proposed model on a broader class of brain tumors. A further future development concerns the construction of models that are able to classify MRI images by also including coronal and sagittal projections. Finally, we would like to define a feature selection method that permits the improvement of models' performances according to correlations of data resulting from several layers of an artificial neural network.

Acknowledgements

We thank Nicola Salerno, Federica Amato, and Alfredo Di Nuccio for supporting the research goals of this work and for their contributions that greatly improved the manuscript.

References

[1] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella. The 'k'in k-fold cross validation. In *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 441–446. i6doc. com publ, 2012.

[2] N. M. Balasooriya and R. D. Nawarathna. A sophisticated convolutional neural network model for brain tumor classification. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–5. IEEE, 2017.

[3] A. Baldo, A. Cuzzocrea, E. Fadda, and P. G. Bringas. Financial forecasting via deep-learning and machine-learning tools over two-dimensional objects transformed from time series. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 550–563. Springer, 2021.

[4] I. Bifulco and S. Cirillo. Discovery multiple data structures in big data through global optimization and clustering methods. In *IV*, pages 117–121, 2018.

[5] B. Breve, L. Caruccio, S. Cirillo, V. Deufemia, and G. Polese. Visual ECG analysis in real-world scenarios. In S. Chang, editor, *The 27th International DMS Conference on Visualization and Visual Languages, DMSVIVA 2021, KSIR Virtual Conference Center, USA, June 29-30, 2021*, pages 46–54. KSI Research Inc., 2021.

[6] P. Chang, J. Grinband, B. Weinberg, M. Bardis, M. Khy, G. Cadena, M.-Y. Su, S. Cha, C. Filippi, D. Bota, et al. Deep-learning convolutional neural networks accurately classify genetic mutations in gliomas. *American Journal of Neuroradiology*, 39(7):1201–1207, 2018.

[7] E.-S. A. El-Dahshan, H. M. Mohsen, K. Revett, and A.-B. M. Salem. Computer-aided diagnosis of human brain tumor through mri: A survey and a new algorithm. *Expert systems with Applications*, 41(11):5526–5545, 2014.

[8] S. B. Gaikwad and M. S. Joshi. Brain tumor classification using principal component analysis and probabilistic neural

network. *International Journal of Computer Applications*, 120(3), 2015.

[9] P. Ghosal, L. Nandanwar, S. Kanchan, A. Bhadra, J. Chakraborty, and D. Nandi. Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network. In *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, pages 1–6. IEEE, 2019.

[10] N. Gordillo, E. Montseny, and P. Sobrevilla. State of the art survey on mri brain tumor segmentation. *Magnetic resonance imaging*, 31(8):1426–1438, 2013.

[11] Y. Gusev, K. Bhuvaneshwar, L. Song, J.-C. Zenklusen, H. Fine, and S. Madhavan. The REMBRANDT study, a large collection of genomic data from brain cancer patients. *Scientific data*, 5:180158, 2018. https://wiki.cancerimagingarchive.net/display/Public/REMBRANDT.

[12] S. Haykin. *Neural networks*, volume 2. Prentice hall New York, 1994.

[13] D. M. Joshi, N. Rana, and V. Misra. Classification of brain cancer using artificial neural network. In *2010 2nd International Conference on Electronic Computer Technology*, pages 112–116. IEEE, 2010.

[14] K. D. Kharat, P. P. Kulkarni, and M. Nagori. Brain tumor classification using neural network based methods. *International Journal of Computer Science and Informatics*, 1(4):2231–5292, 2012.

[15] S. Kumar, C. Dabas, and S. Godara. Classification of brain mri tumor images: a hybrid approach. *Procedia computer science*, 122:510–517, 2017.

[16] R. Leece, J. Xu, Q. T. Ostrom, Y. Chen, C. Kruchko, and J. S. Barnholtz-Sloan. Global incidence of malignant brain and other central nervous system tumors by histology, 2003–2007. *Neuro-Oncology*, 19(11):1553–1564, 2017.

[17] S. Moccia, L. Romeo, L. Migliorelli, E. Frontoni, and P. Zingaretti. Supervised cnn strategies for optical image segmentation and classification in interventional medicine. In *Deep Learners and Deep Learner Descriptors for Medical Applications*, pages 213–236. Springer, 2020.

[18] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem. Classification using deep learning neural networks for brain tumors. *Future Computing and Informatics Journal*, 3(1):68–71, 2018.

[19] L. Nagasai, V. Sriprasath, V. SajithVariyar, V. Sowmya, K. Aniketh, T. Sarath, and K. Soman. Electric vehicle steering design and automated control using cnn and reinforcement learning. In *Soft Computing and Signal Processing*, pages 513–523. Springer, 2021.

[20] J. Naik and S. Patel. Tumor detection and classification using decision tree in brain mri. *International Journal of Computer Science and Network Security (IJCSNS)*, 14(6):87, 2014.

[21] R. Nalbalwar, U. Majhi, R. Patil, and S. Gonge. Detection of brain tumor by using ANN. *image*, 2(3):7, 2014.

[22] O. Neethu and K. Shruti. A reliable method for brain tumor detection using cnn technique. *IOSR Journal of Electrical and Electronics Engineering*, pages 64–68, 2017.

[23] M. F. Othman and M. A. M. Basri. Probabilistic neural network for brain tumor classification. In *2011 Second International Conference on Intelligent Systems, Modelling and Simulation*, pages 136–138. IEEE, 2011.

[24] N. Rani and S. Vashisth. Brain tumor detection and classification with feed forward back-prop neural network. *International Journal of Computer Applications*, 975:8887, 2017.

[25] L. O. Rojas-Perez and J. Martinez-Carranza. Deeppilot: A cnn for autonomous drone racing. *Sensors*, 20(16):4524, 2020.

[26] L. Shen and T. Anderson. Multimodal brain MRI tumor segmentation via convolutional neural networks, 2017.

[27] S. Suhag and L. Saini. Automatic detection of brain tumor by image processing in matlab. In *SARC-IRF International Conference*, 2015.

[28] N. Sumitra and R. K. Saxena. Brain tumor classification using back propagation neural network. *International Journal of Image, Graphics and Signal Processing*, 5(2):45, 2013.

[29] A. Tiwari, S. Srivastava, and M. Pant. Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019. *Pattern Recognition Letters*, 131:244–260, 2020.

[30] R. Xu and D. C. Wunsch. Clustering algorithms in biomedical research: a review. *IEEE reviews in biomedical engineering*, 3:120–154, 2010.

DrawSE2: an application for the visual definition of visual languages using the local context-based visual language specification

Gennaro Costagliola, Mattia De Rosa, Vittorio Fuccella, Vincenzo Raia
Dipartimento di Informatica, University of Salerno
Via Giovanni Paolo II, 84084 Fisciano (SA), Italy
{gencos, matderosa, vfuccella}@unisa.it

Abstract

We present DrawSE2, a new web application that allows the definition of visual languages in a more visual way (according to the local context-based visual language specification). The tool allows the user to create visual language elements, define their attaching areas (the hotspots through which language elements can be connected), and define how they can be linked together to form admissible language sentences. The tool also allows semantic attributes to be defined and enables semantic translation (e.g., to a textual representation). The visual language thus defined can then be used in a diagram editor that allows to draw visual sentences of the language, check their correctness and get their semantic translation.

Keywords: *visual languages, local contex, webapp.*

1. Introduction

Visual languages have been used as part of systems that use visual representations to facilitate communication. Visual sentences include diagrams, maps, images, and pictures, which are used to communicate mental concepts that require spatial settings to be described appropriately. Their purpose is to make it easier for people to communicate, since, when done correctly, visual communication is more direct and instantaneous than spoken or text communication.

This is why visual languages can be found in a variety of contexts, from art to engineering. However, if they are badly designed, they can be difficult to interpret and compose, defeating their purpose. This, for example, may occur when a language has many syntactic rules that bind elements that can be far apart in a sentence. For example, in textual programming languages, one might consider the matching parenthesis in languages such as C or Java.

One way to overcome the problem of syntax dependencies between far language elements is to add shape information to each element, like in the Scratch block visual language. This reduces the composition of a visual sentence (a program in this case) to the creation of a puzzle, with a very simple syntactic rule: "a visual program is syntactically correct if and only if each block (tile) well interlocks with its neighbors". The local shape constraints on each block in this case guarantee the correctness of the whole visual program, regardless of how many elements it is made of.

This is also why block languages are now very popular for teaching introductory programming to non-experts, as well as for prototyping and scripting purposes [23].

We have previously shown [5–10] that many well-known and widely used visual languages (such as unstructured flowcharts, data flow languages, and entity-relationship diagrams) can be syntactically specified mostly using local constraints, rather than complex grammars. This simplifies the design of visual programming languages from a syntactic perspective.

Our methodology, known as *local context-based visual language specification*, only requires the language designer to define the *local context* of each symbol of the language. The local context of a symbol is the set of attributes that define the local constraints that need to be considered for the correct use of the symbol and are the interface that a symbol exposes to the rest of the sentence.

We also defined a way to do a semantic translation of a visual language based on the local context. In particular, we use XPath-like expressions to define the semantic translation rules for the language. These expressions allow us to specify rules for each single language element, rather than defining semantic rules for complete phrases. For a given node in the abstract syntax graph returned by the syntactic phase, we can use these expressions to gather values from its neighbors to be used in the translation. The translation is then expressed by writing simple source code that prints these values.

DOI reference number: 10.18293/DMSVIVA22-014

Although defining a visual language using the local context methodology does not require writing a grammar (which can be quite complex even for the most experienced users), defining the language can still be time-consuming and identifying at first glance the relationships between the various components of the visual language can be difficult.

For this reason in this paper we propose a new tool, a web application called DrawSE2, that allows the definition of visual languages in an almost visual way. The user can create the visual elements of the language (symbols and connectors) by putting together predefined shapes, then define their attaching areas (the hot spots on which symbols and connectors can be attached). The so prepared language element can be positioned on a canvas and visually related by adding placeholders on their attaching areas. This allows the user to define in a simple way and with immediate visual feedback which symbols can be linked together and which cannot.

The tool also offers the possibility to specify attributes such as the number of admissible occurrences of a symbol or how many times an attaching area can be used through contextual menus or panels. If a semantic translation is required (e.g. to a text representation), the tool also offers the possibility of defining the semantic specification by using a tabular interface.

The visual language so defined can then be used to instantiate a diagram editor that allows one to draw sentences of the language, verify their correctness, and get the semantic translation.

The paper is organized as follows: Section 2 describes previous work in this field; Section 3 describes the local context-based visual language specification; Sections 4 and 5 describe our design and DrawSE2, respectively. Finally, Section 6 concludes the paper with a discussion on future work.

2. Related Work

In the past years significant research has been done regarding visual languages and their applications to different scenarios [4,11–13]. Moreover, several strategies have been developed to model diagrams as visual languages sentences. A diagram has been represented either as a set of attributed symbols with typed attributes representing the "position" of the symbol in the sentence (*attribute-based approach*) [17], or a set of relations on symbols (*relation-based approach*) [25]. The two approaches may look different, but both consider a diagram as a set of symbols and relationships between them, that is, a spatial-relationship graph [2] built by adding a node for each graphical symbol and an edge for each spatial relationship between them.

In contrast to the relationship-based approach, where relationships are explicitly represented, the attribute-based approach requires the relationships to be derived from the attribute (equal) values.

Based on these representations, various formalisms have been proposed to represent the syntax of a visual language, each associated with custom scanning and parsing techniques, e.g. (Extended) Positional Grammars [16], Reserved Graph Grammars [30], Constrained Set Grammars [21], Relational Grammars [27] (for other approaches and details see [14] and [22]). In general, such visual grammars are defined by specifying an alphabet of graphical symbols together with their "visual" appearance, a set of spatial relationships generally defined on symbol position and attaching points/areas, and a set of grammar rules, usually in a context-free like format even though their descriptive power is mostly context sensitive.

A large number of tools exist for prototyping visual languages. These are based on different types of visual grammar formalisms and include, among others, VLDesk [15], DiaGen [24], GenGed [1], Penguin [3], VisPro [31], AToM3 [20], VL-Eli [19] and its improvement DEViL [26], and tools dedicated to 3D visual languages such as [28, 29]. However, our *local context-based visual language specification* [10] goes a step further by completely removing the grammar specification.

Despite the fact that context-free rules are well known, it is not easy to define and read visual grammars. This may explain why these technologies have failed to move from laboratories to real-world applications. Many visual languages used today are syntactically simple languages that focus on basic graphic elements and their expressiveness, so there is no need to specify complex grammatical rules.

3. Local Context Specification of Visual Languages

The local context-based visual language specification allows the definition of a visual language both syntactically and semantically, and also enables the definition of a semantic translation of the visual language sentences (e.g., in text format). Its main feature is that it does not make use of grammars in order to allow easier specification of languages. It has been successfully applied to the definition of various visual real word languages, showing that often there is no need for a grammar definition.

A full description of the methodology can be found in [10]. For reasons of space, it is not possible to describe it here in detail, so we will only indicate here its main features while referring to [10] for a complete definition and examples.

According to the local context specification, a visual language is a set of visual sentences on an alphabet of *symbols* and *connectors* (i.e. *language elements*). Each of them is characterized by the following attributes:

- a unique name;
- a graphical appearance;
- the minimum and/or maximum numbers of admissible occurrences in any sentence of the language;
- one or more attaching areas. Each area is characterized by a *unique name*, its *shape* and *location* on the symbol or connector, a set of *local constraints*, such as the number of possible connections to the area (referred to as *connectNum*), and a *type* used to force legal connections among symbol and connector attaching areas. In fact, a connector area can be attached to a symbol area only if they have the same type.
- a number of symbol level constraints involving more than one attaching area;

Moreover, in order to allow a meaningful visual language translation, *textual attaching areas* are possible, i.e. attaching areas that are designed to only contain text. In this case, the local constraints define instead the set of admissible values for the text (e.g. through a regular expression).

The syntactic analysis uses this information (local to the symbol/connector) to check the correctness of the language and produce a graph called abstract sentence graph. This contains a node for each symbol/connector and an edge between all the symbol-connector pairs that are connected.

The Local Context-based Semantic Definition (LCSD) allows the semantic translation of a visual language. The LCSD consists of a sequence of *semantic rules* for each element of the language. Each rule either calculates a *property* or executes an *action*. The properties are calculated through *procedures* making use of XPath-like expressions and possibly validated through a *post-condition*. An action depends on properties and attributes.

Through the post-conditions, an LCSD may better refine the syntactic structure of the language sentences; through the actions, it provides a translation of the sentences. The semantic analysis algorithm uses a data flow model of execution in order to run the semantic translation rules specified for each language element (as opposed to defining semantic rules for complete phrases). In particular, the XPath-like expressions are exploited to gather values from the element neighbors (and use them in the translation). The new methodology was in part implemented as part of the LoCoMoTiVe tool [7]. In contrast to DrawSE2, it is a desktop application that allows the definition of the visual language only partially through the GUI. It also does not allow a visual representation of how language elements can be linked. In designing DrasSE2, we, therefore, sought to overcome these limitations.

4. Designing a visual representation of visual languages

In designing DrawSE2 we had not only the goal of creating a tool that would allow defining (through the local context technique) a visual language in a simple way, but that would also make the created definition easy to understand. To achieve this we decided that the best way was for the definition to be as much as possible a visual language itself.

To this end, we decided that the visual elements comprising the visual language should simply be placed in the language definition (in the x/y position that the author deems most appropriate). Having shown the language elements, the next most important thing is how the elements of the language can be related to each other in an admissible way. The local context definition uses types (designer-defined names) associated with attaching areas to indicate that a connection is permissible between areas with the same type.

This type of information can be trivially visually displayed by a hyperedge (as it connects two or more attack areas together). In our early designs, we tried some of the typical ways to visually represent a hyperedge, but we noticed that, for example in Euler-style visualization, as the number of language elements increases it becomes increasingly visually heavy and difficult to understand (and difficult for the user to draw). For this reason, we decided to use numbered placeholders (colored circles with a number inside them) to indicate that attaching areas with the same placeholder represent allowable connections, as shown in Figure 1.

Although it would be easy to show other information in this type of visualization, such as the number of permissible occurrences for a language element or the number of permissible connections for an attaching area (simply by showing the related text next to the relevant element), we decided to exclude such information from the immediately accessible visual representation because it made it too visually heavy, and decided that such information should be visible only after selecting a specific language element.

These design decisions were then included in DrawSE2.

5. DrawSE2

DrawSE2 is an application that allows visual language definition according to the local context specification in a visual/GUI way, making it easy to define how symbols can be linked together and minimizing the amount of code to be written. This is expected to make the definition of a visual language easier and more understandable even for users with minimal knowledge of grammars and programming. DrawSE2 is based on *diagrams.net* [18], a web application that allows diagrams to be composed using predefined sym-

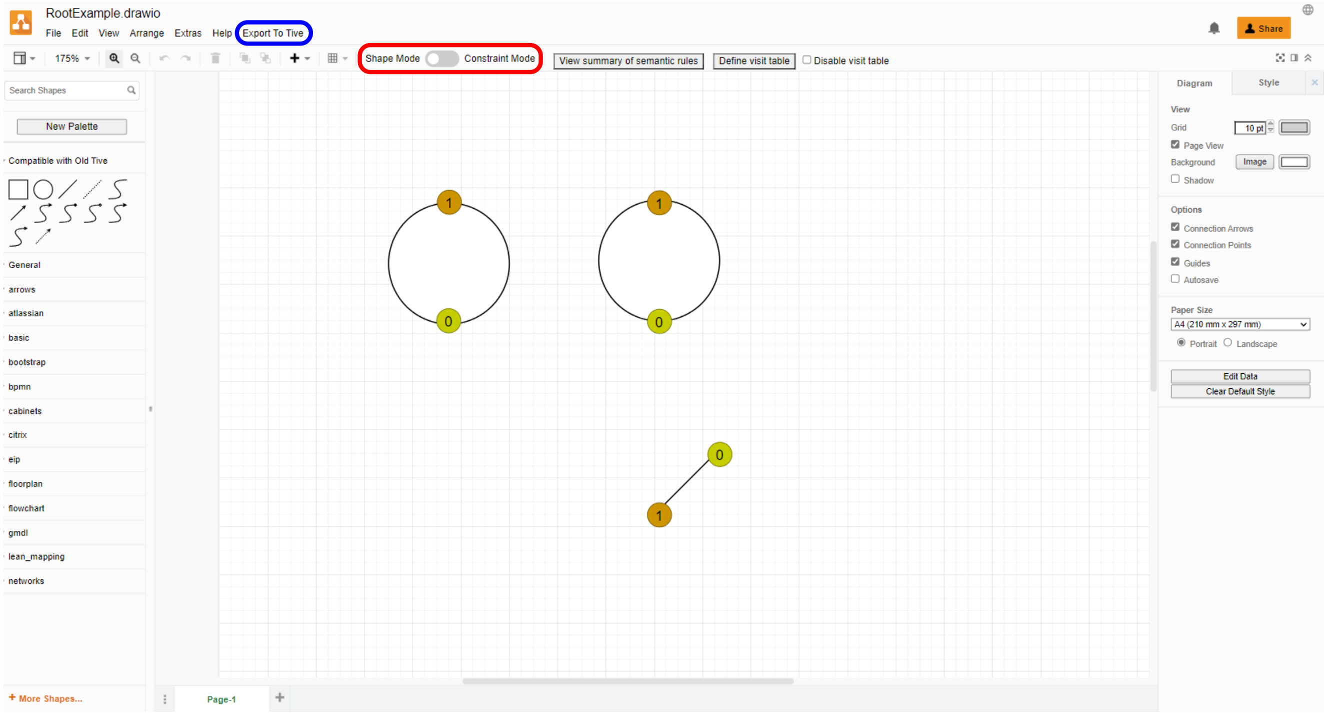
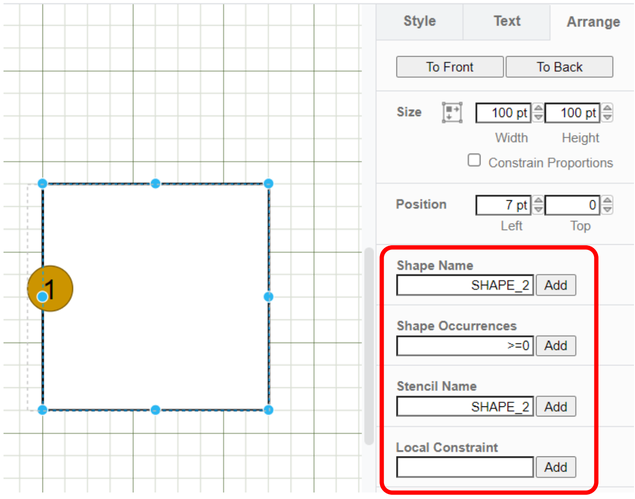


Figure 1: DrawSE2 in shape mode.

bols and connectors, but does not include syntactic/semantic analysis capabilities.

DrawSE2 has two modes: the shape mode in which one can define visual language elements (symbols and connectors, including the composition of predefined elements), element attributes, and semantic rules; and the constraint mode in which one can define the attaching areas for the language elements and the admissible connections between them. It is also possible to export the created language, i.e., to run an instance of an editor (also based on *diagrams.net*) in which it is possible to draw diagrams of the newly defined language and use the syntactic/semantic analysis functions to check for correctness and obtain the semantic translation (e.g., into text) of the drawn diagram.

Figure 1 shows a screenshot of DrawSE2 in shape mode. It is possible to switch to constraint mode and vice versa using the corresponding switch (highlighted in the red box): this changes the display of the canvas contents and the side menu. There are also menus and buttons offering features typical of graphic editors (and already included a *diagrams.net*) such as zoom, undo/redo, for changing properties and styles of graphic elements, etc., which are then available both in shape mode and in constrain mode. Finally, it is possible to export the language using the “Export to TiVe” menu (highlighted in the blue box): this causes the editor to open in a new browser tab. In the next sections, we will describe each mode in detail.



Size: 100 pt (Width), 100 pt (Height)

Position: 7 pt (Left), 0 (Top)

Shape Name: SHAPE_2

Shape Occurrences: >=0

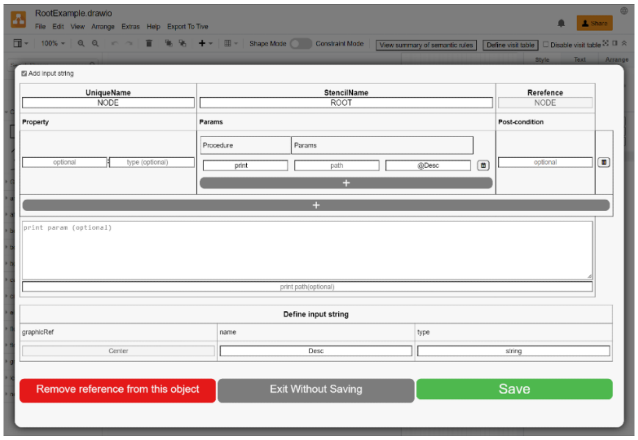
Stencil Name: SHAPE_2

Local Constraint:

Figure 2: Defining the attributes of an element.

5.1. Shape Mode

The Shape mode allows one to define the symbols and connectors used in the language. For a symbol or connector to become part of the language, it is sufficient for it to be placed on the canvas by dragging it from the panel on the left, which contains a set of predefined symbols and connectors. One can also create their own custom symbols by putting together several predefined elements. Such



☑ Add input string

UniqueName	StencilName	Rerefence
NODE	ROOT	NODE

Property	Params	Post-condition
optional : type (optional)	Procedure · Params print · path · @Desc	optional

print param (optional)

print path(optional)

Define input string

graphicRef	name	type
Center	Desc	string

Figure 3: Table for defining semantic rules.

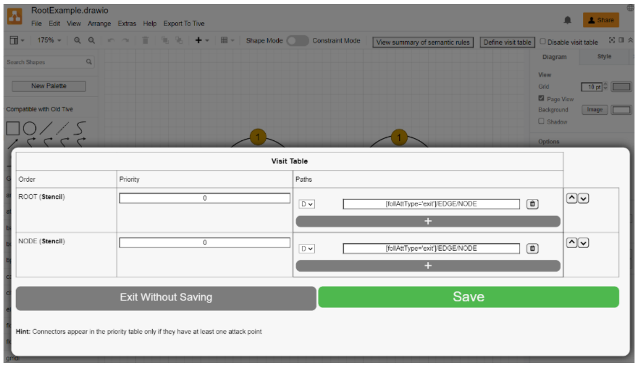
predefined elements also include lines and curves so there is considerable flexibility, considering also that it is possible to import graphic elements in png format. This is accomplished by selecting individual elements and using the group function found in the context menu (accessible by right-clicking).

By selecting a symbol it is also possible to define the following attributes (from the local context methodology) thanks to the panel on the right: name; number of permissible occurrences in a language sentence; expanded name (shown to language users), local constraints involving multiple attaching areas (more on this in the next section). An example is shown in Figure 2.

Also in this mode, semantic rules can be defined. To perform this action there is a "Define semantic rules" option in the context menu accessible for each element. In this case, an editable table will be shown that will allow these rules to be defined congruently with the local context methodology.

In particular, the table, as shown in Figure 3, allows defining the list of symbol properties and how these are to be computed. There is also a text area in which the code to produce the semantic translation of that language element can be entered. For convenience, there is the listing of the textual attaching areas of the symbol, if any (since they are likely to be referenced in the code that produces the semantic translation, since they may contain user-written text). Finally, there are buttons to save, close, or remove the table altogether.

The "Define visit table" button at the top of the GUI allows one to define the visit table, which is used to define the order in which language elements will be visited during semantic analysis/translation. In particular, it is possible to define the order and priority of each element, and the paths associated with them. Again, there are buttons to save and close, as shown in Figure 4.



Visit Table

Order	Priority	Paths
ROOT (**Stencil**)	0	D [follAttType='exit']/EDGE/NODE
NODE (**Stencil**)	0	D [follAttType='exit']/EDGE/NODE

Hint: Connectors appear in the priority table only if they have at least one attack point

Figure 4: Example of a visit table definition.

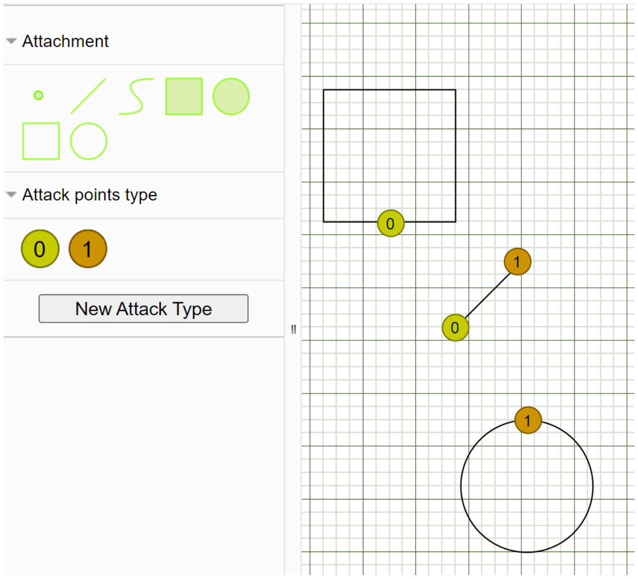
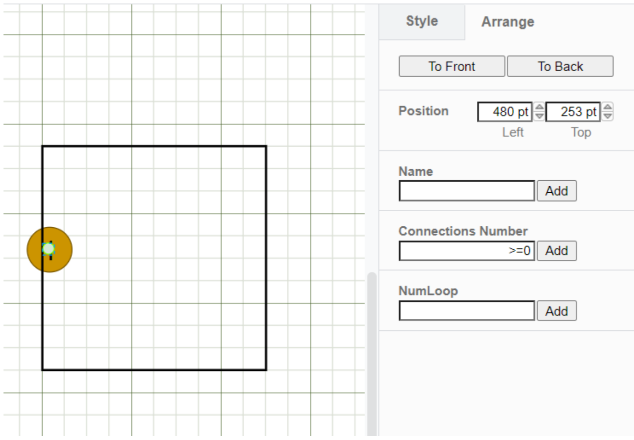


Figure 5: Example of using "New Attack Types".



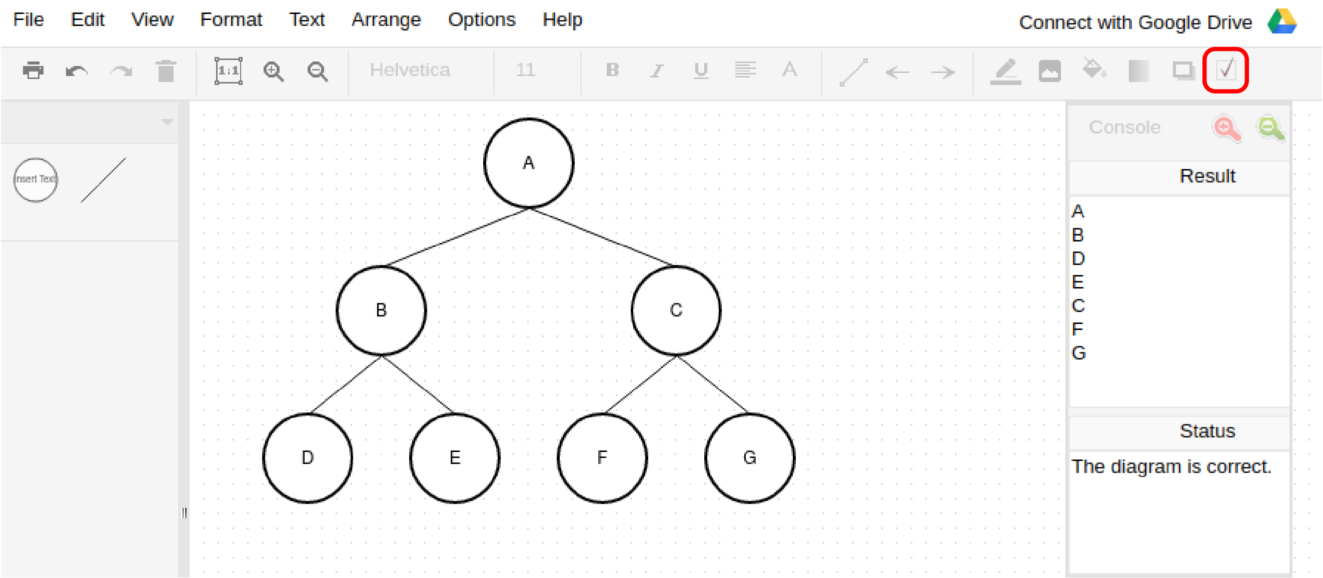
Position 480 pt Left 253 pt Top

Name

Connections Number >=0

NumLoop

Figure 6: Defining the properties of an attaching area on a symbol.



Result

A
B
D
E
C
F
G

Status

The diagram is correct.

Figure 7: Example of editor created for the tree language. The button highlighted in the red box launches the correctness check and semantic translation. The semantic translation (a preorder visit) is shown on the right panel.

5.2. Constraint Mode

The Constraint Mode allows one to define the attaching areas for each symbol or connector in a visual way. When switching to constraint mode, the language elements remain visible but become uneditable, and the side panel on the left instead shows the set of attaching areas ("Attachment" panel in Figure 5) that one can drag onto the language elements in the canvas in order to add them.

It is also possible to define how language elements can be connected to each other. This can be done by creating placeholders (through the "New Attack Types" button). They are represented graphically by circles of different colors and containing numbers. Placeholders with the same number can be placed on the attaching areas of language elements in order to indicate that these elements can be connected together in a valid visual sentence, as shown in Figure 5.

It is also possible to define properties (from the local context methodology) for each attack type, including: name (optional: useful if one needs to reference it in the semantic specification); maximum number of elements that can be attached to it; limits on self-loops (connector leaving and arriving on the same area), as shown in Figure 6.

5.3. Visual language editor: TiVe

The TiVe editor uses the visual language definition and is also based on *diagrams.net*. It differs from it in that it shows in the left panel only the symbols and connectors that are part of the language, and that it adds a button that allows the user to perform the correctness check of what has been drawn. If successful, the semantic translation (if defined) or a message confirming correctness will be shown, or an error message otherwise. Figure 7 shows an example of this editor.

6. Conclusions and further works

We presented DrawSE2, a new tool for defining visual languages in a more visual way (according to local context methodology). The user can create the visual elements of the language, define their attaching areas, and define how they may connect with each other by placing placeholders on their attaching areas. The tool also allows the user to define the semantic attributes and the semantic translation through tabular GUI.

Future work will involve empirical evaluation of the software by performing user studies involving both experts in visual languages and users with no experience in the field. Regarding the latter users, it is planned to involve computer science students studying compilers (of textual programming languages). After this evaluation, the software will be further refined according to the received feedback and by correcting any remaining bugs.

7. Acknowledgment

The authors thank Gerardo Brescia and Vincenzo Caputo for their support in implementing DrawSE's earlier versions.

References

[1] R. Bardohl. Genged: a generic graphical editor for visual languages based on algebraic graph grammars. In *Visual Languages, 1998. Proceedings. 1998 IEEE Symposium on*, pages 48–55, Sep 1998.

[2] R. Bardohl, M. Minas, G. Taentzer, and A. Schürr. Handbook of graph grammars and computing by graph transformation. chapter Application of Graph Transformation to Visual Languages, pages 105–180. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1999.

[3] S. S. Chok and K. Marriott. Automatic construction of intelligent diagram editors. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, pages 185–194, New York, NY, USA, 1998. ACM.

[4] G. Costagliola, M. De Rosa, A. Fish, V. Fuccella, R. Saleh, and S. Swartwood. Knotsketch: A tool for knot diagram sketching, encoding and re-generation. In *The 22nd International Conference on Distributed Multimedia Systems*, pages 16–25, 2016.

[5] G. Costagliola, M. De Rosa, and V. Fuccella. Local context-based recognition of sketched diagrams. *Journal of Visual Languages & Computing*, 25(6):955–962, 2014.

[6] G. Costagliola, M. De Rosa, and V. Fuccella. Local context-based recognition of sketched diagrams. In *The 20th International Conference on Distributed Multimedia Systems*, pages 321–328. Knowledge Systems Institute, August 2014.

[7] G. Costagliola, M. De Rosa, and V. Fuccella. Extending local context-based specifications of visual languages. *Journal of Visual Languages & Computing*, 31, Part B:184 – 195, 2015.

[8] G. Costagliola, M. De Rosa, and V. Fuccella. Fast prototyping of visual languages using local context-based specifications. In A. Guercio, editor, *The 21st International Conference on Distributed Multimedia Systems*, pages 14–22. Knowledge Systems Institute, August 2015.

[9] G. Costagliola, M. De Rosa, and V. Fuccella. Fast prototyping of visual languages using local context-based specifications. *J. Vis. Lang. Sentient Syst.*, 1, 2015.

[10] G. Costagliola, M. De Rosa, and V. Fuccella. Using the local context for the definition and implementation of visual languages. *Comput. Lang. Syst. Struct.*, 54:20–38, 2018.

[11] G. Costagliola, M. De Rosa, V. Fuccella, and M. Minas. Visual exploration of visual parser execution. *Multimedia Tools and Applications*, 81(1):299–317, 2022.

[12] G. Costagliola, M. De Rosa, V. Fuccella, and S. Perna. Visual languages: A graphical review. *Information Visualization*, 17(4):335–350, 2018.

[13] G. Costagliola, M. De Rosa, and M. Minas. Visual parsing and parser visualization. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 243–247, 2019.

[14] G. Costagliola, V. Deufemia, and G. Polese. A framework for modeling and implementing visual notations with applications to software engineering. *ACM Trans. Softw. Eng. Methodol.*, 13(4):431–487, Oct. 2004.

[15] G. Costagliola, V. Deufemia, and G. Polese. Visual language implementation through standard compiler–compiler techniques. *Journal of Visual Languages & Computing*, 18(2):165 – 226, 2007.

[16] G. Costagliola and G. Polese. Extended positional grammars. In *Proceeding 2000 IEEE International Symposium on Visual Languages*, pages 103–110, 2000.

[17] E. J. Golin. Parsing visual languages with picture layout grammars. *J. Vis. Lang. Comput.*, 2(4):371–393, Dec. 1991.

[18] JGraph Ltd. diagrams.net. https://www.diagrams.net, 2022.

[19] U. Kastens and C. Schmidt. Vl-eli: A generator for visual languages - system demonstration. *Electr. Notes Theor. Comput. Sci.*, 65(3):139–143, 2002.

[20] J. d. Lara and H. Vangheluwe. Atom3: A tool for multi-formalism and meta-modelling. In *Fundamental Approaches to Software Engineering*, pages 174–188, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[21] K. Marriott. Parsing visual languages with constraint multiset grammars. In *Programming Languages: Implementations, Logics and Programs*, pages 24–25, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[22] K. Marriott and B. Meyer. On the classification of visual languages by grammar hierarchies. *Journal of Visual Languages & Computing*, 8(4):375 – 402, 1997.

[23] Y. Matsuzawa, Y. Tanaka, and S. Sakai. Measuring an impact of block-based language in introductory programming. In *SaITE 2016 - Stakeholders and Information Technology in Education - IFIP Advances in Information and Communication Technology*, volume 493, pages 16–27. Springer, Cham, 2016.

[24] M. Minas and G. Viehstaedt. Diagen: A generator for diagram editors providing direct manipulation and execution of diagrams. In *Proceedings of the 11th International IEEE Symposium on Visual Languages*, VL '95, pages 203–, Washington, DC, USA, 1995. IEEE Computer Society.

[25] J. Rekers and A. Schurr. A graph based framework for the implementation of visual environments. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 148–155, Sep 1996.

[26] C. Schmidt, U. Kastens, and B. Cramer. Using devil for implementation of domain-specific visual languages. In *Proceedings of the Workshop on Domain-Specific Program Development*, page 38, 2006.

[27] L. Weitzman and K. Wittenburg. Relational grammars for interactive design. In *Visual Languages, 1993., Proceedings 1993 IEEE Symposium on*, pages 4–11, Aug 1993.

[28] J. Wolter. Devil3d - A generator framework for three-dimensional visual languages. In *Proceedings of the 18th International Conference on Distributed Multimedia Systems, DMS 2012, August 9-11, 2012, Eden Roc Renaissance, Miami Beach, FL, USA*, pages 171–176. Knowledge Systems Institute, 2012.

[29] J. Wolter. Specifying generic depictions of language constructs for 3d visual languages. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 139–142, 2013.

[30] D.-Q. Zhang and K. Zhang. Reserved graph grammar: a specification tool for diagrammatic vpls. In *Proceedings. 1997 IEEE Symposium on Visual Languages (Cat. No.97TB100180)*, pages 284–291, 1997.

[31] D.-Q. Zhang and K. Zhang. Vispro: a visual language generation toolset. In *Proceedings. 1998 IEEE Symposium on Visual Languages (Cat. No.98TB100254)*, pages 195–202, 1998.

Indicators in *Super Mario Maker 2*: Evolution and “Oral” Tradition in Visual Languages

Nathan W. Eloe[1], Trevor C. Meyer[2]
School of Computer Science and Information Systems[1]
Department of Language, Literature, and Writing[2]
Northwest Missouri State University, USA
{nathane,tmeyer}@nwmissouri.edu

Abstract

As a key element of user experience; communicating expectations to a user through visual elements instead of natural language can produce a more intuitive interface as users interact with a system. While such visual languages are developed by domain experts for specific purposes, these languages can also grow, change, and evolve within a community of users. Video games are one area where communication with the user directly affects the enjoyability, usability, and accessibility of the system. While work has been done to create and leverage visual languages in order to gamify learning or improve accessibility, this research focuses on the creation of these visual languages by external experts. This exploration of a “crowd-sourced” context-aware visual language examines a system of indicators that has evolved over time, created by the very users of the language, to communicate the expectations and necessary actions to complete a task with other members of the community. Investigating such a language, how its changed and used, can teach designers of visual languages ways to more effectively use them in their own work.

Index terms— Visual Languages, Video Games, Usability and Accessibility

1 Introduction

Video games have long motivated progress in many aspects of Computer Science; algorithms (e.g. The Fast Inverse Square Root [2], often attributed to John Carmack in the implementation of *Quake*), hardware, and even education and pedagogy [1, 8] are but some areas of Computer Science that have been improved through video games. Games fulfill a particular need in entertainment; they can cater to a diverse audience while providing an interactive, not passive, entertainment that can be consumed by one or more players. Unlike storytelling, art, or video content alone, video games allow the player to interact with a tactile environment while processing audio and visual information, creating a multisensory experience. While these multisensory experiences need to abide by normative assumptions, as consumer products, increased awareness of the need for accessibility has motivated developers to accommodate sensory, learning, and affective differences.

Video games that present the player with a goal and expect them to reach it provide a challenge for developers: how do they communicate to the player the actions required to complete the posed goal? Often these games are presented as a series of levels (like the *Super Mario* series), though some use a more open world (ranging from gated progression “Metroidvania” games to fully open world games like *Skyrim*). In level-based games, each level must essentially act as an independent implementation of a visual language, providing cues for the player to determine the necessary series of actions to reach goal.

Initially, professional level designers carefully crafted these levels. The progression of technology and the availability of tools and knowledge has allowed amateur designers to create levels within the confines of a given game’s mechanics to challenge their friends, create unique puzzles, or randomize the elements of a game to allow players a unique experience in the game every time they play.

Indeed, anyone with the desire to learn the appropriate flavor of assembly can modify games like Super Mario World and create fully-fledged “ROM hacks”; collections of custom levels, and sometimes custom mechanics, that can be played by others. One particular type of ROM hack, often called a Kaizo ROM hack

DOI reference number: 10.18293/DMSVIVA22-003

(or Kaizo game, named for one of the earliest hacks in this category: Kaizo Mario), focuses on tricks and mechanics that require near perfect execution that can be achieved through immense of skill and/or practice. These ROM hacks afford a sufficiently skilled creator an incredible amount of flexibility with mechanics and game assets; almost anything can be accomplished or communicated given the right assembly code.

In 2015, Nintendo released *Super Mario Maker* for the Wii U, which simplified the creation of new Mario levels. This was followed on the Nintendo Switch based *Super Mario Maker 2* in 2019. These games allowed anyone with an internet connection to create levels using a predefined palette of sprites, enemies, powerups, and game themes and make them available for anyone else with a copy of the game to play. It did not take long for players of Kaizo hacks to begin to create their own levels, and over time they developed a body of incredibly challenging levels with creative setups using the limited palette of tools available in the Mario Maker games. These creators (also players of these games) encountered the same question professional level designers do for games developed by game studios: "how do I tell the player what they need to do?"

In ROM hacks, players can insert arbitrary text and assets, allowing near limitless ways to communicate intent and requirements; however, this flexibility is not afforded to level creators in the Mario Maker games. There is a limited ability to create letters using combinations of sprites and objects in the game, and a brief description can be attached to each level, but this often doesn't provide enough space to convey the necessary information.

Instead the creators developed a simple, semi context-aware visual language using the simple primitives provided by the game itself. This language has no glossary or dictionary, or even perhaps a written description; it instead relies on the skill of the player to understand the actions available to them to determine what should be done next. Interestingly, it may be almost impossible to track down the first use of these indicators; levels sometimes get removed for a variety of reasons and as such the genesis of some of these indicators and any textual description they might have had may be lost.

Over time the language has even evolved to account for updates to the game that have added new abilities and graphical elements. This evolution does not seem to be communicated explicitly between creators; instead a creator sees and understands the use of an indicator in a level, and then uses it in their own (perhaps in a slightly different manner). This language develops similarly to oral tradition; players "hear" a story (see the use of an indicator) and incorporate it into their own levels, as an allusion, homage, or even outright theft! The understanding of these visual lexemes is left to the level "reader". This language has essentially been crowd-sourced, leading to a common understanding (and indeed at times to setups in levels that are so common they no longer require indicators to be understood).

The primary purpose of this paper is to investigate elements of this language of indicators using subset of Green's Cognitive Dimensions of Notations [6] as a starting point and the lessons that can be learned from analyzing a visual language where the creators and consumers are the same group. Some language elements are particularly interesting with respect to the Cognitive Dimensions. A deeper investigation of this and similar languages, perhaps involving other analysis tools such as Moody's Physics of Notations [9] and drawing connections to linguistics, is left for future work.

To accomplish this task, we will provide an introduction to the relevant limitations in the level creator in Mario Maker 2. After analyzing some individual elements of the language of indicators this paper will proceed into a few full screen examples demonstrating how the language can be read in the context of a level. Definitions of relevant player abilities will be provided as necessary. Finally an analysis of some of the lessons visual language developers can take from such a language that has naturally evolved will be presented.

2 Related Work and Background

2.1 Visual Language and Level Design in Mario Games

In an interview [3], Shigeru Miyamoto discusses the design of the classic Level 1-1 of the original *Super Mario Brothers* (*SMB1*). In this interview he discusses not only how the level was designed last, but also how it trains the player to play the game through a process of rewards and punishments. On the first screen alone, if the player just walks (or doesn't move at all) the Goomba walking across the screen will cause the player's death. If the player moves, avoids the Goomba, and jumps into the classic question mark block, they are rewarded with a coin. By putting the player in situations where they can escape danger or be rewarded by experimenting, the design team trains the player to perform certain actions simply by introducing basic elements that will be multiplied and complicated as the player progresses. In this way, the player learns that question mark blocks are positive, and that they must jump on certain enemies to defeat them. However deep pits should be avoided, as should being touched by enemies as these will lead to Mario's demise (or at least a loss of a powerup). The careful training of players over time into the "official" Mario game-level language is a foundational part of the language of indicators that has evolved over time. The following assumptions will be made about the language that Mario level designers have trained players to understand since *SMB1*:

- Coins are rewards, and they should be collected (motivating the player to move Mario to collect them)
- Question mark blocks, disguised question mark blocks, and turn blocks can be struck from below by the player (or from the side using a thrown item which is introduced), which often rewards the player with a power-up or some other form of level progression
- Enemies can be safely jumped on unless they have some form of protection (like spikes or fire); later games gave Mario the ability to safely jump on these enemies using a spin jump move.

The core mechanics of the game are exposed to the player through low-stakes experimentation in the first few moments of the first level, not explicitly through text. This is truly a marvel of game design principles. The mechanics and visual elements from *Mario* games are the foundation for level creation in the *Super Mario Maker* games.

2.2 Limitations in Mario Maker 2

The way *Mario Maker 2* conveys information has two major limitations. First each screen of the level is a grid that is 24 "blocks" wide by 13 or 14 blocks tall. This space limitation limits what knowledge can be conveyed through simple "pixel art". Additionally, the only free-form text that can be distributed with levels comes from the level description which has a limited character count. Players can leave small comments in the level, but they may not always be trustworthy or helpful (as they do not come from the level creator).

3 A Language of Indicators

Building on the previous assumptions (and referencing them), this paper will focus on three extensions to the visual language: the P-Block, a C-shaped track, and a curved track. These indicators communicate modifications to position, action, and movement, respectively. All images below were captured from the author's console using the screenshot feature and edited as necessary

3.1 Character "Target": The P-Block

In *Super Mario Maker 1* and earlier updates for *Super Mario Maker 2*, creators were limited to using the coin to indicate to players where they should attempt to send their character. Due to the limited palette, the coin was a multi-purpose reward indicator, promising progression if the player performs the correct action at that location. Level creators were afforded a new option with a later update to the game: the P-Block (Figure 1), which occupies a single grid space on the screen.

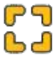


Figure 1: The P-Block

Mechanically, when this block is in the state shown in Figure 1 it behaves like a background object the character can pass through. If the character hits a P-Switch, the block becomes a solid object. The background state of these blocks makes it useful as an indicator, but particularly clever creators can use it as an indicator in the background state and as a platform or wall in its activated state.

This indicator then acts as a general "aim here" message to the player, and is used in many of the situations the coin was in earlier levels (allowing the coin to be used as a different kind of indicator). It is especially useful when the item or enemy player will be landing on is not yet apparent but will be by the time Mario reaches that location, assuming all preconditions have been met. Interestingly, this indicator's intended meaning can change based not only on how it is used but also the style used to create the level.

When a creator chooses a level style based on *Super Mario World*, *New Super Mario Bros. U*, or *Super Mario 3D World*, Mario has access to his Spin Jump move; the interaction between Mario and objects he lands on often changes based which jump was used. This requires the player to be aware of both what they will be landing on and the skills available in the chosen game style.

Visually, the negative space in this indicator in its inactive state somewhat resembles a cross-hair or target, which evokes the real world idea of aiming for a particular location (an example of Green's Closeness of Mapping). While the color cannot be changed by the level creator, extra meaning can be conveyed through placement or number of indicators (which is related to Green's Secondary Notation and Escape from Formalism). For example placing one P-Block directly above another often indicates that the player should land in that location twice. An additional example is provided in the partial level read in Section 4.

This indicator can be confusing if there are multiple reachable from the player's current location, making Premature Commitment particularly a concern at times, though often after one or two mistakes it becomes more clear as to which location should be the player's next target.

3.2 Throw an Item: Shaped Tracks

In all level styles except *SMB1*, the player has the ability to pick up, carry, and throw objects. These sprites can interact with the world, triggering switches, breaking blocks, or just providing an object the player can land on later. Kaizo levels often require objects to be thrown precisely to allow the player to proceed. As quite flexible game-objects, tracks can be placed in relatively arbitrary shapes, but must either be a closed loop or a simple path. Often, a closed square loop in a 3x3 grid is used to indicate to the player that they should wait at a location (or something will appear there). Alternately, Tracks can be placed in the shape of a "Z" to indicate that the player should press the Z button on the controller (this is a fairly explicit use of indicators to communicate intent to the player). While arrows pointing in various directions can be placed in levels, creators often rely on a ⊏, ⊐ or ⊔ shaped set of tracks (Figure 2).

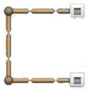


Figure 2: Tracks shaped to indicate the direction to throw a held item

When one side of the closed square loop is omitted, it tells the player to throw or drop the item they are currently holding in the direction of the missing edge when they are inside the square. Usually if the player releases or throws the correct held object in the direction indicated by the opening while Mario is in the outlined box the item will end up exactly where it needs to be for the next required steps in the level to "just work".

This indicator is interesting with respect to the Cognitive Dimensions of Notation. The size of the symbol is larger than the corresponding arrow (3x3 instead of roughly 1x3), but provides additional context of the player's location when the object should be thrown. Choosing to trade terseness for additional information may depend on the available space. Arrows are also frequently used to indicate a direction the player should go if it's not immediately clear; this may be a reason why the shaped track was chosen.

Additionally, this particular indicator has a hidden dependency: the player must be holding an object for it to have meaning, and if there are multiple objects available they need to be holding the correct item. If a player reaches this indicator without an object, it may be a hint that a previous part of the notation has been misinterpreted or the player just did not execute the previous parts of the level perfectly.

Also, there is nothing obvious about this particular indicator that indicates throwing. The closest it comes is indicating a position and a direction. Closeness of Mapping may be this indicator's weakest Cognitive Dimension, though there is not a particular symbol available that would necessarily directly communicate the idea of "throw" to the player.

One Cognitive Dimension this notation rates particularly well at is Consistency. The orientation of the notation (which side is open) indicates the direction the player should send the held object. Once a player has learned that the ⊏ shaped track means "throw the held object right", the player can then infer that ⊔ shape means throw the object up (in game styles that support that) and a ⊐ shaped track indicates that the object should be thrown backwards.

3.3 Twirl Jump: Curved Track

Super Mario Maker 2 allows an additional trick with Tracks: a Track placed diagonally can curve instead of going in a straight line (as in Figure 3). Unlike the track configuration in Figure 2 that encodes its meaning in the empty space contained within the track and its missing side, the path this indicator expresses its meaning in its shape.

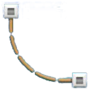
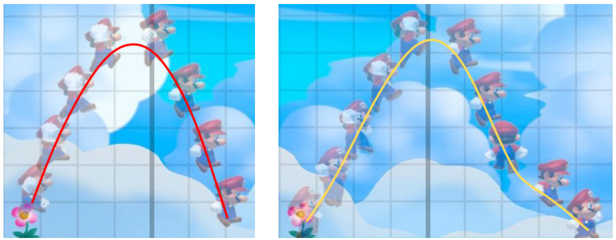


Figure 3: Tracks curved to indicate when to twirl

In the New Super Mario Bros. U. and Super Mario 3D World level styles the player can perform a maneuver called an air twirl. This briefly stalls the character's downward momentum; when performed during a jump this effectively extends the reach of the jump slightly as in Figure 4.



(a) Jump without twirl **(b)** Jump with twirl

Figure 4: A comparison of Mario's jumps with and without an air twirl

Note the curve in the path at the end of the jump in Figure 4b.Mario's path is almost exactly the shape of the curved Track, an excellent example of Closeness of Mapping. Treating the grid as a standard Euclidean graph, Mario begins the air twirl at $(6, 2)$ which stalls his downward momentum as he travels to $(7, 1)$; those are exactly the grid spaces the track would occupy; beginning an air twirl at the top square of the track will cause Mario to follow the path indicated by the track very closely. A player who understands the mechanics afforded by the level style can read this indicator and map that to the necessary controls to make the character follow the path shown by the notation.

4 Reading the Level: An Example

Figure 5 contains an annotated representation of the first few screens of the level Mechanical Manacles by creator Donkeymint (Level Code 8QH-JRX-GLG). This image was created by splicing live game play images captured directly on the Nintendo Switch, and annotated to explain how a player who knows the mechanics of both the game and level style can complete the level with enough skill or practice.

The player begins on the left side of the screen. The actions that must be taken do have a strong ordering, and it is not immediately obvious how to get from the starting location to the first safe platform (the coins at indicator I3). Because the timing of the actions is also critical, skilled level creators will sometimes add "reset doors" as seen at the beginning of the level to give players time to analyze the situation or retry a particularly difficult trick, which helps avoid premature commitment. The indicators themselves do not necessarily have a way to indicate order of actions, often the only way to determine what action to take next is to look at what the only reachable indicator is.

While there is no immediately clear path forward, the indicator I1 tells the player that they should land there. Because this level is in the style of Super Mario World there are two choices of how to get there: a regular jump or a spin jump; one of these jumps will lead to success while the other will cause the player to fall into the Piranha Plant (the bottom enemy in the stack) or the saw blades below where they will die. Since there are only two options and this is the first jump in the level, failure is not too punishing; however a player who understands the mechanics of the game can glean more information from the indicator.

I1 is placed to the left of the top enemy, called a Mechakoopa, and if the player was simply using it as a way to bounce higher, then the player would land directly on top of the enemy. However, not every enemy reacts to jumps in the same way; the player needs either learn through experience or research how enemies interact with different player actions. The placement of I1 indicates some directionality in the desired end result. With a regular jump, the Mechakoopa would collapse and fall straight down (or be knocked slightly to the side), but a spin jump will send this particular enemy in a direction opposite to the side they were hit from (if they were hit from the left, they would move right, and vice versa) [4].

By using a spin jump on the Mechakoopa at I1, the enemy will be stunned and sent to the right, landing on or near the indicator at I2 (depending on the exact timing of the jump). Because the character continues spinning they can land on the stunned Mechakoopa and safely make it to the coins at I3. At this point, the next indicator (I4) tells the player that they should throw something to the right, but currently has nothing in hand to throw. Depending on the exact timing of these first two jumps, one of two things will happen. If the player landed again on the left side of the stunned Mechakoopa it will be knocked to the right where it will land on the note block and bounced up to the top of the icicles (directly next to the player). But if the player doesn't knock the enemy to the right, it will eventually recover and begin walking around, turn around at the one way gate to the left of I2, and then walk to the note block and bounce up to the player waiting at I3. At this point the Mechakoopa can be picked up (or re-stunned and then picked up), giving the player something to throw.

After jumping and throwing the enemy to the right at I4, there is only one safe place to land: the checkered platform below the indicator. However, this platform will start to fall quickly, so the player will need to quickly decide what to do. The next indicator (I5) tells the player to jump there, but at this point there is nothing to land on. However, when the Mechakoopa is thrown from I4, it lands on top of the note block to its right, which spawns a flying enemy directly at

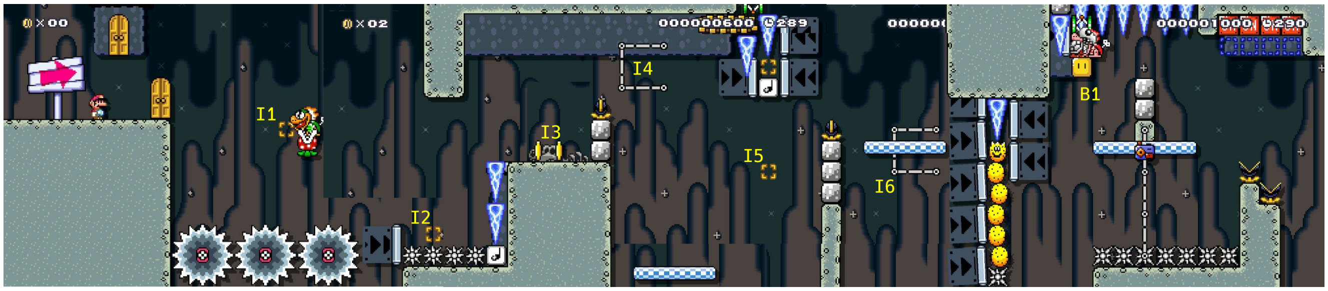


Figure 5: A partial level with indicators. Footage of a player reading and completing this level can be found at [5]

I5. The player can then bounce off of this enemy and land on the checkered platform that is in the middle of indicator I6.

Once again, the player is told to throw an object from a location but isn't holding anything. The mechanism above the note block above I5 causes the Mechakoopa (which was thrown at I4) to bounce up and then over, landing on the checkered platform. This enemy should then be thrown at I6; there are P-Block indicators behind the tall yellow enemy (a Pokey) that can be seen as the enemy moves. The Mechakoopa destroys the Pokey and falls to land on the spike at the bottom of the screen. A spin jump will allow the player to jump off of the stunned Mechakoopa and land on the right most checkered platform, which then begins to move up and down.

At this point the creator forgoes the use of custom indicators and relies on the visual language established over time by the developers of the *Super Mario Brothers* series. After landing on the right most platform (which begins to move), the player has exactly one choice left: hit the turn block B1. This triggers the mechanism above the turn block, which sends another stunned Mechakoopa to land on the right side of the checkered platform. At this point the player again has exactly one choice to live: pick up the stunned Mechakoopa and throw it up into the On/Off blocks which are hidden by the timer in the upper right corner of Figure 5. Hitting these blocks will cause blue blocks to become solid and red blocks to become background objects, allowing the player to progress further in the level.

An important consideration for this level and the spliced image is illustrated in Figure 5. The images were captured by a player who has not devoted hours of practice to develop the skill that professionals will hone over time. In fact, the amount of skill required to beat the level is independent of the knowledge and ability to translate the language; despite being a mediocre (at best) player, I was able to progress as far in the level as I was through repetition and practice; the intended action was always clear, even if my muscle memory and skill was not developed enough to allow successful execution of every required move.

5 Conclusion

Language designers can take multiple lessons, or reminders perhaps, from investigating such as the one we've shown here. Visual languages are often used to lower the barrier of entry for individuals completing a new task. But experts create mechanisms for communicating succinctly that require a learning curve. For example experts can use shorthand, a glyph-based method of shortening written communication, to increase the speed with which information can be recorded. Quickly decoding shorthand requires practice and knowledge of which system encoded the information. While useful, such methods for communication have a purpose but are not explicitly designed for simplicity or guiding novice users through completing a task.

This language of indicators serves much the same purpose as shorthand: meaning must be conveyed using limited resources in a constrained space. A visual language is used because unrestricted words and lettering are mostly unavailable to level creators. By removing the expectation that the language be immediately accessible to novice players and their knowledge of the game mechanics, the language can be concise and still convey varied meanings with a single indicator based on the context in which it is used.

This is not to say that accessibility is not important; in fact visual languages should continue to be an important part of aiding in accessible design goals. The role of visual languages in wider accessibility is beyond the scope of this paper. This language of indicators enables communication within an audience that shares common knowledge and skills. Understanding the target audience is important when devising any form of communication. Domain specific visual languages should be designed with a specific audience in mind; the intended audience needing certain domain specific knowledge to comprehend the visual language should not be a barrier to the design of the language.

The language of indicators is a prime example of "designing" a language with a specific target audience in mind. How often is the user experience and functionality of a deployed system designed and developed seemingly without consideration for the intended users? Updates resulting in removal of used features or strange UX choices often result in members of the user base reaching out to each other and the developers of the system as in [7]. To quote one reply from that particular thread: "Any time we need to tell instructors 'you have to redesign your course / assessment to fit into the software constraints' you know there is a poor software design." Any visual language must be designed and implemented with the target audience in mind.

However the most intriguing aspect of this language is that it was designed by players for players. Unlike the Hmong Script or the Sequoyah Syllabary, with a single author, many people have built the language over the life cycle of two different Mario Maker games. This language has evolved as updates have changed elements used to indicate meaning. But tracing the origins of specific indicators, like "throw" are impossible because the online services for the original game have been shut down, and levels can (and could) be deleted by both Nintendo and the creator. There was no meeting where a group sat down and decided how that configuration of lines should mean "throw the thing". Perhaps the originator of the indicator put a note in the level description as to what it meant; over time though that indicator has become widespread. A player would play a level, determine the meaning of an indicator, and then use it in their own creations. If that creator's understanding of the indicator was slightly different, they might use it in a different way. As such, the meaning and usage of this language has evolved in much the same way stories and history are passed down in oral tradition.

Language and communication evolve; the meanings of words and images take on new meaning over time as the concepts that are being represented change and grow. The role of visual languages in enabling accessible and intuitive experiences for all users has been increasing, and will continue to evolve with language as a whole. As visual language designers it is important to keep the intended audience firmly in mind when building an experience and a mechanism for guiding users through completing a task. However, the intended audience might be a collection of skilled experts who've developed ways to communicate succinctly with each other. This language of indicators serves as a fun reminder couched in a video game that intention can be communicated using limited tools if the audience's knowledge and skill is leveraged. It can serve as a reminder to UI and UX designers that perhaps they should look at how members of the target audience are already conveying information to each other and common knowledge within the intended users. That may give the language designer a starting point to provide an experience that is familiar,more intuitive, and more comfortable to the user while leveraging their experience and knowledge to enhance how information is communicated. This can also ease the burden on the designer, as much of the design work may already have been completed as the users allowed their communication to change, allowing more time for implementation, testing, and refinement.

References

[1] D. Bitonto, T. Roselli, V. Rossano, E. Frezza, and E. Piccinno. An game to learn type 1 diabetes management. *Proceedings: DMS 2012 - 18th International Conference on Distributed Multimedia Systems*, pages 139–143, 01 2012.

[2] J. Blinn. Floating-point tricks. *IEEE Computer Graphics and Applications*, 17(4):80–84, 1997.

[3] Eurogamer. Miyamoto on World 1-1: How Nintendo made Mario's most iconic level. `https://www.youtube.com/watch?v=zRGRJRUWafY`.

[4] FANDOM. Mechakoopa | Super Mario Maker Wiki | Fandom. `https://supermariomaker2.fandom.com/wiki/Mechakoopa`, Dec. 2020. Accessed: 2022-03-02.

[5] GrandPOOBear. RACE this arrow from MAX MAP HEIGHT [SUPER MARIO MAKER 2]. `https://www.youtube.com/watch?v=5wQlFpLYVsg`, Nov. 2021. Accessed: 2022-03-02.

[6] T. R. Green. Cognitive dimensions of notations. *People and computers V*, pages 443–460, 1989.

[7] jlubkinchavez. Solved: New quizzes survey workaround and complete/incomplete. `https://community.canvaslms.com/t5/New-Quizzes-Users/New-Quizzes-survey-workaround-and-complete-incomplete/td-p/194757`, June 2020. Accessed: 2022-03-09.

[8] E. J. Marchiori, Ángel del Blanco, J. Torrente, I. Martinez-Ortiz, and B. Fernández-Manjón. A visual language for the creation of narrative educational games. *Journal of Visual Languages and Computing*, 22(6):443–452, 2011.

[9] D. Moody. The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.

Integration of SMGA and Maude to Facilitate Characteristic Conjecture

Dang Duy Bui, Duong Dinh Tran, Kazuhiro Ogata
School of Information Science
Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
Email: {bddang,duongtd,ogata}@jaist.ac.jp

Adrián Riesco
Facultad de Informática
Universidad Complutense de Madrid
José García Santesmases, 9 Ciudad Universitaria, 28040
Madrid, Spain
Email: ariesco@ucm.es

***Abstract*—SMGA is a visualization tool that focuses on helping human users to conjecture characteristics of a protocol. Those characteristics can be lemma candidates to prove that the protocol enjoys its desired properties in theorem proving. In previous work, interaction has been indirectly shown as one promising approach. Hence, it is worth focusing on interaction in SMGA. In the present paper, we revise SMGA to provide interactive features that can assist human users in conjecturing characteristics. Furthermore, we integrate SMGA and Maude, a high-performance reflective language and system so that the revised version of SMGA can use some powerful features of Maude, such as parsing, reachability analysis, and model checking. We conduct a case study to demonstrate the usefulness of these features by showing in detail the use of each feature in conjecturing characteristics.**

***Keywords*-graphical animations; Maude; SMGA; state machine; interactive features**

I. Introduction

State Machine Graphical Animation (SMGA) [1] is a tool to visualize protocols based on state machines formalizing the protocols. The main purpose of SMGA is to make humans able to conjecture characteristics of the protocols that can be used as lemma candidates to prove that the protocols enjoy some desired properties in theorem proving. The tool uses visual information because visual perception is one of the human strength [2]. Interaction is a substantial aspect of visualization tools from which humans can get insights via interacting [3]. Some case studies of SMGA [4], [5], [6] indirectly showed that interaction was one promising factor to find non-trivial characteristics. Therefore, in the present paper, we aim to revise SMGA to provide new and revised interactive features for human users to conjecture characteristics. In the paper, the revised version of SMGA is called as r-SMGA for short.

The input of SMGA is a state picture template and a state sequence. The state picture template is designed by human users and it is an important part in SMGA [5]. The output of SMGA is graphical animations of the state sequence based on the state picture template. By observing such animations, human users can conjecture some characteristics. In SMGA, there are two features: control and pattern matching. Human users can use some functions of the control feature, such as play and run step, to control animations. Given a state sequence, the pattern matching feature can help users to find states satisfying some conditions via regular expressions. In r-SMGA, we integrate SMGA and Maude [7], a high-performance reflective language and system so that r-SMGA can use some powerful features of Maude, such as parsing, reachability analysis, and model checking. Then, the pattern matching feature is revised to be able to handle associative-commutative pattern matching that cannot be handled by using regular expressions. Moreover, while observing the animations, users can use some interactive features to focus or hide elements in which users are interested or less interested, respectively. We also provide some visualization for commonly used data structures, such as queue and array.

We conduct a case study in which the Suzuki-Kasami distributed mutual exclusion protocol is used to demonstrate the usefulness of the new and revised features for conjecturing characteristics. In r-SMGA, we first design a state picture template and use a new feature to visualize queue and array data structures. Based on the animations, some characteristics of the Suzuki-Kasami protocol are conjectured by using our new and revised features, and some tips proposed by Bui and Ogata [5]. Those characteristics are then confirmed by the search command of Maude by using r-SMGA.

The rest of the paper is organized as follows. Sect. II mentions some preliminaries such as state machines, Maude, and SMGA. Sect. III introduces the Suzuki-Kasami protocol and its specification. In Sect. IV, we describe the ideas of the new and revised features in r-SMGA. To demonstrate the usefulness of r-SMGA, some characteristics of the Suzuki-Kasami protocol are guessed and confirmed by using the new and revised features in Sect. V. Sect. VI discusses some related work. Finally, we conclude the present paper in Sect. VII.

II. Preliminaries

A. State Machines and Maude

A state machine $M \triangleq \langle S, I, T \rangle$ consists of a set S of states, a set $I \subseteq S$ of initial states, and a binary relation $T \subseteq S \times S$ over states. $(s, s') \in T$ is called a state transition

DOI reference number: 10.18293/DMSVIVA22-006

and may be written as $s \rightarrow_M s'$. The set $R \subseteq S$ of reachable states with respect to M is inductively defined as follows: (1) for each $s \in I$, $s \in R$ and (2) for each $(s, s') \in T$, if $s \in R$, then $s' \in R$. A state predicate p is an invariant property w.r.t. M if and only if $p(s)$ holds for all $s \in R$. A finite sequence $s_0, \ldots, s_i, s_{i+1}, \ldots, s_n$ of states is called a finite computation of M if $s_0 \in I$ and $(s_i, s_{i+1}) \in T$ for each $i = 0, \ldots, n-1$.

In this paper, to express a state of S, we use a braced associative-commutative collection of name-value pairs. Associative-commutative collections are called soups, and name-value pairs are called observable components. That is, a state is expressed as a braced soup of observable components. The juxtaposition operator is used as the constructor of soups. Suppose $oc1, oc2, oc3$ are observable components, and then $oc1$ $oc2$ $oc3$ is the soup of those three observable components. A state can be expressed as $\{oc1\ oc2\ oc3\}$. There are many possible ways to specify state transitions. In the present paper, Maude [7], a programming/specification language based on rewriting logic is used as one candidate to specify state transitions as rewrite rules. Maude can specify complex systems flexibly and is also equipped with several formal analysis techiniques, such as reachability analysis and LTL model checking. A rewrite rule starts with the keyword `rl`, followed by a label enclosed by square brackets and a colon, two patterns (terms that may contain variables) connected with =>, and ends with a full stop. A conditional one starts with the keyword `crl` and has a condition following the keyword `if` before a full stop. The following is the form of a conditional rewrite rule:

`crl` $[lb]$: l => r `if` ... /\ c_i /\ ...

where lb is a label and c_i is a part of the condition, which may be an equation $lc_i = rc_i$. The negation of $lc_i = rc_i$ could be written as $(lc_i =/= rc_i) = $ `true`, where = `true` could be omitted. If the condition ... /\ c_i /\ ... holds under some substitution σ, $\sigma(l)$ can be replaced with $\sigma(r)$.

Maude provides the `search` command that allows users to find a reachable state from t such that the state matches the pattern p and satisfies the condition c:

`search [n,m] in MOD :` t =>* p `such that` c `.`

where `MOD` is the name of the Maude module specifying the state machine, `n` and `m` are optional arguments stating a bound on the number of desired solutions and the maximum depth of the search, respectively. `n` typically is 1 and t typically represents an initial state of the state machine.

Maude provides LTL model checking so that we can check whether a system satisfies a desired property, which is expressed as an LTL formula. Maude can check the system that starts from `init` satisfies φ by the following command:

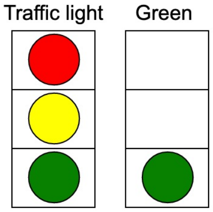
`reduce modelCheck(init,` φ`) .`

Maude returns true if the system satisfies φ. Otherwise, a counterexample is returned.

B. State Machine Graphical Animation (SMGA)

State Machine Graphical Animation (SMGA) was originally developed by Nguyen and Ogata [1]. The main purpose of SMGA is to make humans able to conjecture characteristics of protocols. Such characteristics can be used as lemma candidates to prove that the protocols hold some properties in theorem proving. We divide SMGA into two phases: preparation and control as shown in Fig. 1. In the preparation phase, SMGA requires a state picture template (called a state picture design in [5]) and a state sequence as the input. The state picture template is designed by users while the state sequence is generated by Maude from a formal specification of a protocol. Designing the state picture template is an important part of SMGA [5] because if the state picture template is simple, such as it contains texts only, then, it is boring and hard to observe characteristics of protocols [4]. Based on the input, SMGA produces graphical animations as the output. Observing such animations allows human users to conjecture characteristics. In the control phase, users can control the animations, such as changing the speed of the animations (for running automatically), running step by step in which each step can be regarded as a state transition. In addition, given a state sequence input, SMGA allows us to search for some states in the state sequence such that such states satisfy some conditions. This feature uses regular expressions to conduct the search, and is called Find Patterns in [4], [5].

SMGA basically provides two kinds of visualization for an observable component: (1) text display and (2) analogous display. (1) presents a value of an observable component as text while (2) presents a value of an observable component followed by what users expect, such as visual elements. For example, an observable component simulates a traffic light that contains one of three values: Red, Yellow, and Green. The following figure displays a state picture template (on the left-hand side) and a state picture when the value of the traffic light is Green (on the right-hand side). In the figure, the text on the top is displayed as (1) while three circles are displayed as (2).



III. Suzuki-Kasami Distributed Mutual Exclusion Protocol

A. Description

The Suzuki-Kasami distributed mutual exclusion protocol (also known as the Suzuki-Kasami protocol) was proposed

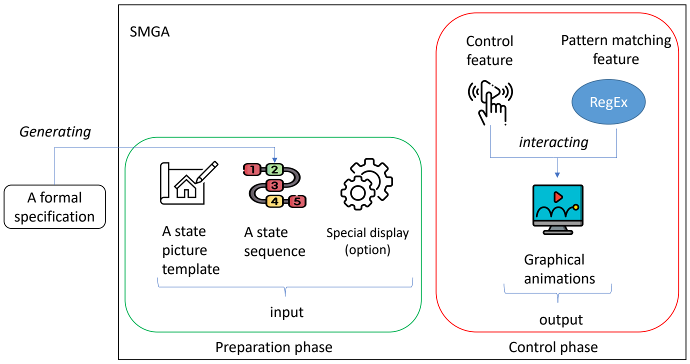


Figure 1. An overview of SMGA

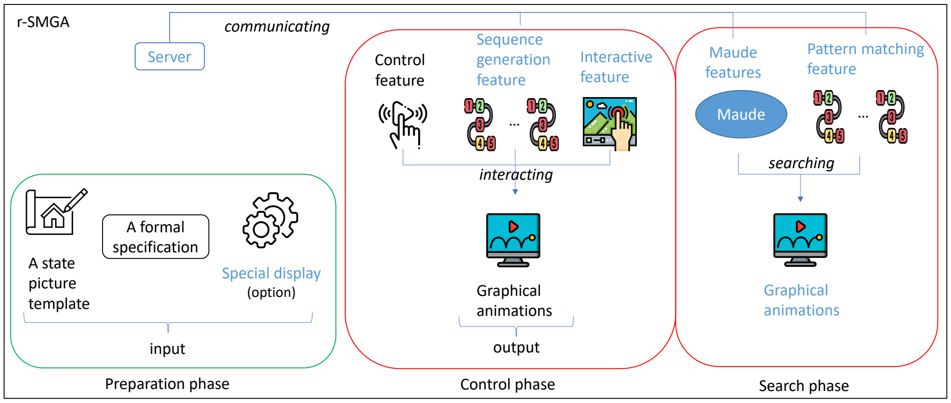


Figure 2. An overview of r-SMGA

by Suzuki and Kasami [8]. In the protocol, if a node owns a privilege, then the node can enter the critical section. The privilege can be transferred to other nodes in the network. In the protocol, there are N nodes participate and $1, \ldots, \mathrm{N}$ are used for their identifiers. Node is defined as a set of all node's ids $\{1, \ldots, \mathrm{N}\}$. Each node can communicate with each other by exchanging messages in the network. In the protocol, there are two kinds of messages named $\mathrm{request}$ and $\mathrm{privilege}$. A $\mathrm{request}$ message is defined as $\mathrm{request(j,n)}$, where j is the id of a node that sends the message and n is a number of request - a natural number. A $\mathrm{privilege}$ message is defined as $\mathrm{privilege(q,\ a)}$, where q is a queue of node's ids and a is an array of natural numbers whose size is N. In the protocol, there are two procedures P1 and P2 used for each node $\mathrm{i} \in \mathrm{Node}$ and the two procedures are described in Fig.3.

$\mathrm{requesting}$ and $\mathrm{have\_privilege}$ are two Boolean variables. $\mathrm{requesting}$ is true if node i wants to enter the critical section, otherwise it is false. $\mathrm{have\_privilege}$ is true if node i owns the privilege, otherwise it is false. queue is a queue of Node containing node's ids that are requesting to enter the critical section. ln and rn are arrays of natural numbers whose size is N. $\mathrm{ln[j]}$ for each node $\mathrm{j} \in \mathrm{Node}$ is the number of node j's request grant most recently. rn records the largest request number received from each of the other nodes. For each node i, its rn is always meaningful while its queue and ln are meaningful when node i owns the privilege. For each node $\mathrm{i} \in \mathrm{Node}$, initially, $\mathrm{requesting}$ is false, $\mathrm{have\_privilege}$ is true if $\mathrm{i} = 1$, otherwise it is false, queue is empty and each element of ln, and rn is 0.

Procedure P1 is used for node i if it wants to enter the critical section. First, the node sets $\mathrm{requesting}$ to true. If node i owns the privilege, it moves to the critical section. Otherwise, it increments $\mathrm{rn[i]}$ and transfers the $\mathrm{request}$ message $\mathrm{request(i, rn[i])}$ to other nodes. Then, node i waits to receive the privilege and set $\mathrm{have\_privilege}$ to true if it receives the privilege. It moves to the critical section after that. Once node i enters the critical section, it updates $\mathrm{ln[i]}$ by $\mathrm{rn[i]}$. queue is updated by checking queue and i's ln with identifiers of other nodes. After that, if queue is empty, it means that node i still keeps the privilege, sets $\mathrm{requesting}$ to false and leaves P1. Otherwise, $\mathrm{have\_privilege}$ is set to false and node i transfers the $\mathrm{privilege}$ message $\mathrm{privilege(deq(queue), ln)}$ to a node which is the top of the queue.

Whenever the request message $\mathrm{request(j, n)}$ is trans-

```
try(i)      ⟷  rem   procedure P1
setReq(i)   ⟷  l1      requesting := true;
chkPrv(i)   ⟷  l2      if ¬have_privilege then
incRN(i)    ⟷  l3        rn[i] := rn[i] + 1;
                         for all j ∈ {1, …, N} – {i} do
sndReq(i)   ⟷  l4          send request(i, rn[i]) to node j;
                         endfor
                         wait until privilege(queue, ln) is received;
wtPrv(i)    ⟷  l5        have_privilege := true;
                       endif
exit(i)     ⟷  cs      Critical Section;
cmpReq(i)   ⟷  l6      ln[i] := rn[i];
                       for all j ∈ {1, …, N} – {i} do
                         if (j ∉ queue) ∧ (rn[j] = ln[j] + 1) then
updQ(i)     ⟷  l7          queue := enq(queue, j);
                         endif
                       endfor
chkQ(i)     ⟷  l8      if queue ≠ empty then
                         have_privilege := false;
trsPrv(i)   ⟷  l9        send privilege(deq(queue), ln) to node top(queue);
                       endif
                       request := false;
rstReq(i)   ⟷  l10   endproc

                     // request(j, n) is received; P2 is indivisible.
                     procedure P2
                       rn[j] := max(rn[j], n);
                       if have_privilege ∧ ¬requesting ∧ (rn[j] = ln[j] + 1)
recReq(i)   ⟷          then have_privilege := false;
                              send privilege(queue, ln) to node j;
                       endif
                     endproc
```

Figure 3. An Algol-like language of the Suzuki-Kasami protocol

ferred to node i, node i runs procedure P2. However, procedure P2 must be atomically executed. First, rn[j] is updated if it is greater than n. Then, by checking have_privilege, requesting, and rn[j] of node i, node i sets its have_privilege to false and sends the privilege message privilege(queue, ln) to node j.

B. A Specification of the Suzuki-Kasami Protocol

We can formalize the Suzuki-Kasami protocol as a state machine in Maude. Nat, Bool, Loc, Queue, and Array are defined as a set of all natural numbers, a set of Boolean values, a set of all locations (e.g., l1 and cs), a set of all queues of Node, and a set of all arrays of natural numbers whose size are N, respectively. The form of a request message is msg(i, req(j, k)), where i ∈ Node is the receiver, j ∈ Node is the sender, k ∈ Nat is a request number, msg is the constructor of messages, and req is the constructor of request. The form of a privilege message is msg(i, priv(q, a)), where i ∈ Node is the receiver, q ∈ Queue is a queue of nodes, a ∈ Array is an array of natural numbers, and priv is the constructor of privileges. The network is formalized as a soup of messages, consisting of request and privilege messages. Let Message be a set of all soups of messages, void ∈ Message denotes an empty network.

To formalize the Suzuki-Kasami protocol, some observable components are used as follows::

- (nw: ms) indicates that the network is ms where ms ∈ Message. Initially, ms is void.
- (queue: q) says that the meaningful queue is q ∈ Queue. Initially, q is empty.
- (ln: a) says that the meaningful ln is a ∈ Array Initially, a is initArray(N) that denotes an array of size N such that each element is 0.

We add two new observable components queue and ln because they are meaningful when some node holds the privilege. Therefore, we use them as a part of each state to visualize the protocol in the next section. Each node i contains the following observable components:

- (pc[i]: l) says that node i is located at l where l ∈ Loc. Initially, l is rem.
- (have_privilege[i]: b) says whether a node owns the privilege. If so b is true; otherwise, it is false. Initially, if i is 1, b is true; otherwise, b is false.
- (requesting[i]:b1) says whether a node wants to enter the critical section. If so, b1 is true; otherwise, it is false. Initially, b1 is false.
- (queue[i] : q1) says that queue[i] is q1 where q1 ∈ Queue. Initially, q1 is empty.
- (rn[i]: a1) says that rn[i] is a1 where a1 ∈ Array. Initially, a1 is initArray(N).
- (ln[i]: a2) says that ln[i] is a2 where a2 ∈ Array. Initially, a2 is initArray(N).
- (idx[i]: j) says that idx[i] is j where j is a natural number used as a loop variable. Initially, j is 1.

To specify the Suzuki-Kasami protocol in Maude, we first divide the protocol into 13 regions as shown in Fig. 3. The name of each region is put on the left side, such as try(i), and exit(i). We suppose that each node is located at one of 12 regions in P1. We require 13 transition rules to specify the 13 regions. Let us explain the rule updQ(i) as follows:

```
rl [updateQueue] :
    (pc[I]: l7) (idx[I]: K) (rn[I]: RN) (ln[I]: LN)
    (queue[I]: Q) (queue: Q)
    => (pc[I]: if K == N then l8 else l7 fi)
       (idx[I]: if K == N then 1 else K + 1 fi)
       (rn[I]: RN) (ln[I]: LN)
       (queue[I]: if K =/= I and not(K \in Q)
                       and (RN[K] == (LN[K]) + 1)
                  then put(Q,K) else Q fi)
       (queue: if K =/= I and not(K \in Q)
                        and (RN[K] == (LN[K]) + 1)
                  then put(Q,K) else Q fi) .
```

where I, K, RN, LN, and Q are Maude variables that belong to Node, Nat, Array, Array, and Queue, respectively. The node I changes to l8 if the index K is equal to N. Otherwise it stays at l7. If K is not equal to I, K does not belong to Q, and RN[K] is equal to LN[K]+1, then K is added to queue[I] and queue. Let us repeat that queue is the same as queue[I] if queue[I] is meaningful and is updated as `queue[I]` is. The other rules work similarly.

IV. Features in r-SMGA

A. Intuitive Idea

The main goal of the present paper is to provide interactive features for helping users to conjecture properties. In r-SMGA, users can interact with the tool by playing elements of the state picture template and get insights by searching some information from the state sequence input. We have integrated r-SMGA and Maude so that r-SMGA can use powerful features of Maude, such as reachability analysis, parsing, and LTL model checking. Moreover, Maude has a rich pattern matching feature constructed by rich grammars, such as context free grammar so that users can search more various information as they want than regular expressions in the previous version. To do that, we use a server as a bridge to communicate between r-SMGA and Maude. The server uses Maude bindings [9] to communicate with Maude via APIs while the server uses sockets to communicate with r-SMGA via message passing. New features are mentioned in the rest of the section.

B. New Features

Fig. 2 displays an overview of r-SMGA in which light-blue texts refer to the new and revised features. In r-SMGA, we divide the tool into three phases: preparation, control, and search. The main purpose of the preparation and control phases is to produce an input and an output while the search phase focuses on analyzing data by searching. In the preparation phase, we use a state picture template and a formal specification of a protocol as the input that is fed into r-SMGA. Note that users do not need to prepare a state sequence as in the previous version. Moreover, we provide a feature called *Special display* to help users to visualize some specific data structures, such as array and queue. Some other visualization features, such as displaying network containing huge messages in [6] can be reused in r-SMGA.

In the control phase, we implement two new features called *Sequence generation* and *Interactive*. The *Sequence generation* feature aims to automatically generate a state sequence based on the formal specification of a protocol on the fly. It consists of five functions:

- **Default generation**: The function automatically generates a random sequence (by selecting randomly one of the successor states for a next state) whose length is up to a fixed number (100 by default). This function guarantees that two consecutive states are different in the sequence. Once a new state sequence is generated, it will be added to a list where its index denotes the state sequence. Users can select any state sequence in the list by an index to reuse without generating it again.
- **Update**: The function uses a selected state sequence from the list, then generates a new random state sequence and replaces the selected state sequence by the new one. Users can adjust the length of a state sequence before producing a new state sequence.
- **Add**: The function works similar as **Default generation** except that users can set a length of a new state sequence before generating.
- **Clear**: The function erases a selected state sequence in the list.
- **Reset list**: The function erases all state sequences in the list.

Users can utilize the *Interactive* feature to interact with a state picture template or a state picture while observing the animations. There are two functions in the feature as follows:

- **Focus**: This function allows users to focus on selected elements in a state picture template or a state picture by displaying only those elements and not displaying the rest of the elements.
- **Hide**: Users can select elements in a state picture template or a state picture that they want to hide. Then, the selected elements are not displayed on the screen.

We also provide three more functions for each function above: undo, redo, and reset. Undo allows users to go to the previous action, while redo allows users to go to the next action that users already did before. Reset allows users to go back to the original state picture template. Note that the function reset is different to **reset list** in the *sequence generation* feature. In addition, users can use the *Interactive* feature while running some other features.

In the search phase, there are two features named *Maude* and *Pattern matching*. In the *Maude* feature, we provide two functions called search command (reachability analyzer) and model checking as follows:

- **Search command**: The input of this function is a specification and a command including parameters mentioned in Sect. II. Given the corresponding parameters, such as an initial state, a target state, and a number of solutions, the function calls to the Maude `search` command. If the number of solutions in the parameters is one, the function returns a path leading to the target state from the initial state. When the number is greater than one, the function returns a list of paths where indices in the list denote the paths. In the list, users can select a number as an index to observe the corresponding path. If there is no solution, an alert message is returned.
- **Model checking**: Given a specification of a protocol, an LTL formula, and an initial state, we can conduct model checking with r-SMGA. If the protocol does not satisfy the formula with the initial state, a counterexample is returned that has the form of a state sequence with a loop. Note that if the reachable state space of the protocol is huge, **Search command** and **Model checking** can stuck or take a long time to return results.

In the *Pattern matching* feature, the purpose is to find states

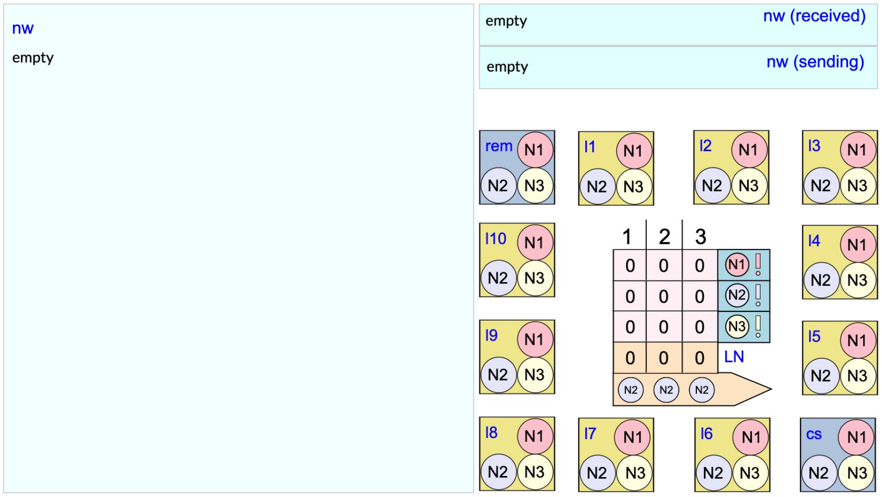


Figure 4. A state picture template of the Suzuki-Kasami protocol.

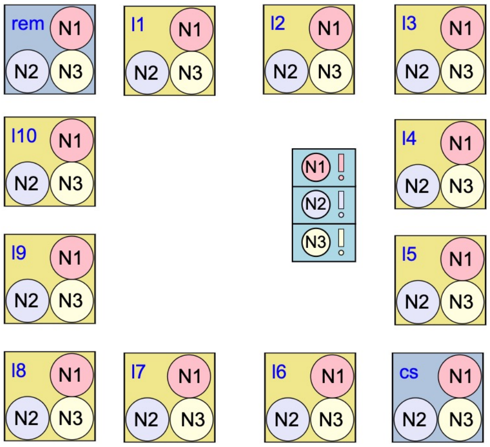
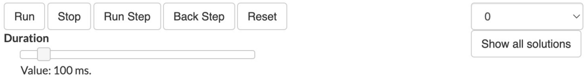


Figure 5. A state picture template of three observable components: pc, privilege, and requesting.

satisfying some conditions from the sequence input. In r-SMGA, the *Pattern matching* feature is implemented based on some features in Maude so that it can work on various cases, such as finding some specific messages in the network that we cannot do with the previous version. This feature uses sequences in the list and a pattern with conditions written in Maude as the input. The pattern and the conditions are defined in a similar way to the target state and the conditions in **Search command**. The output is a list of states that match the pattern with the conditions, or a message "no solution" if no state is matched. Three functions are provided as follows:

- **Pattern matching on a sequence**: Users select a sequence from the list and fill a pattern with conditions. Then, the function returns a list of matched states from the selected sequence.
- **Pattern matching on some sequences**: Users can select some sequences from the list and fill a pattern with conditions. If it is successful, the function returns a list of the selected sequences. Users can select one sequence in the list to observe matched states in such sequence.
- **Pattern matching on all sequences**: The function works in a similar way to **Pattern matching on some sequences** but in this case it is applied to all sequences in the list.

In the search phase, we provide two ways to display the output: (i) a still picture that includes all matched states and (ii) animations. Displaying the output as animations can help human users to recognize the difference between the matched states that may be useful to conjecture characteristics. We will show its usefulness in the next section. The following figure shows functions of the displaying the output as animations.



where buttons on the left side are similar to the functions in the control feature; a list on the right top side is a list of solutions found by *Maude* feature and *Pattern matching* feature; 0 in the list is the index of the first solution of the function **Search command**, or the first sequence in the list found by functions of the *Pattern matching* feature; the button on the right-bottom side is to load all matched states from *Maude* and *Pattern matching* features. We summarize the new and revised features with their purposes and functions in Table. I.

Table I

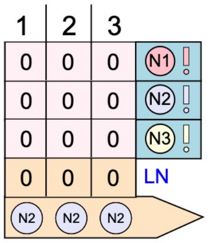
SUMMARY OF NEW AND REVISED FEATURES.

Name of feature	Purpose	Functions
Special display	Visualizing some specific data structures	Array and queue
Sequence generation	Generating a state sequence on the fly and keeping state sequences in a list	Default generation, add, update, clear and reset list
Interactive feature	Interacting with elements from a state picture template or a state picture	Focusing and hiding
Maude feature	Using Maude features that Maude supports	Search command and model checking
Pattern matching feature	Searching states by matching a pattern with conditions	Pattern matching on a sequence, some sequences, all sequences in the list
Displaying the output of the search phase	Displaying the output in the search phase as animations	Control functions for animations and showing all matched states

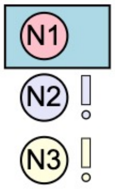
V. EXPERIMENT

A. Applying the New and Revised Features to the Suzuki-Kasami Protocol

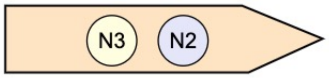
To introduce how to use the new and revised features of r-SMGA and demonstrate its usefulness, we use the Suzuki-Kasami protocol as a case study. Let us suppose that three nodes participate in the protocol. Each node contains some observable components, such as `have_privilege`, `requesting`, and the array `rn`. In the preparation phase, we prepare the state picture template shown in Fig. 4 and the protocol specification. We borrow some visualized techniques in [4] and redesign some observable components such as `have_privilege`, `requesting`, `queue`, `rn`, and `ln`. Based on some tips [5], observable components should be visual as much as possible. Then, in the new state picture template, we redesign `have_privilege` and `requesting` using analogous display. We use the *Special display* feature to visualize `queue`, `rn`, and `ln`. For other observable components, please refer to the work [4]. The following figure shows the revised design:



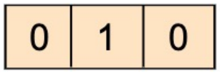
where three blue rectangles and three exclamation marks on the right side represent the `have_privilege` and the `requesting` of the three nodes, respectively. Three circles with different colors inside blue rectangles represent the labels of the three nodes. From top to down, three light-pink rectangles with three numbers inside represent the array `rn` of three nodes 1, 2, and 3, respectively. The numbers 1, 2, and 3 on the top figure represent indices of arrays starting from 1. Light-orange rectangle with three numbers inside and light-orange arrow with circles inside represent the array `ln` and the `queue`, respectively. A blue rectangle and an exclamation mark inside will be displayed if the `have_privilege` and the `requesting` of a node are true. The following figure shows a case in which the `have_privilege` of node 1 is true, nodes 2 and 3 are false; and the `requesting` of nodes 1,2, and 3 are false, true, and true, respectively:



In the previous version, a queue was used with text display only. In r-SMGA, we can make it usable with the analogous display. r-SMGA requires users to design values for each position in a queue. Therefore, for displaying `queue`, there are nine circles in which each position contains three circles representing three nodes. The circles of each position represent the value of the queue and so if the queue is empty, nothing is display. The following figure shows a case in which the queue is `2 | 3 | empty`:



In the previous version, an array was used with text display only. In r-SMGA, users can make an array display as the array structure in conventional programming language. For example, the array `ln` denoted `(1 : 0),(2 : 1),(3 : 0)` is displayed in the following figure. Three arrays `rn` are displayed similarly.



In the control phase, users can generate a state sequence based on the *sequence generation* feature. This action can be repeated to create a list of state sequences. Users can modify or change the specification to generate a new state sequence. Note that, when users modify or change the specification, the old state sequences in the list are still available. Those state sequences should be updated; users can use **Reset list** to erase those state sequences; users can keep them to compare with new state sequences. *Interactive* features and features in the search phase will be discussed in the next sub-sections.

B. Guessing Characteristics Based on the New Features

Bui and Ogata [5] have proposed some tips for guessing characteristics. We summarize the tips and show the usefulness of our new and revised features based on them.

- **CC-T1**: Concentrating on one observable component, users can find some specific values on it from which users can conjecture some characteristics.
- **CC-T2**: Concentrating on two different observable components, users can find a relation between them.
- **CC-T3**: Searching states in which some observable components have some specific values, and concentrating on other observable components on those states, users can find some relations on those observable components.
- **CC-T4**: Investigating conjectured characteristics, users can find some other characteristics.

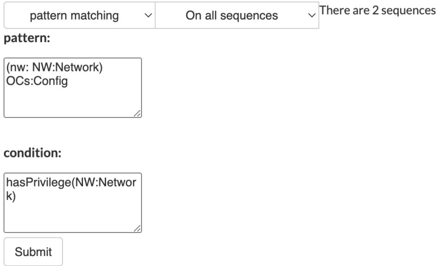
where **CC-T** stands for Characteristic Conjecture Tip. For **CC-T1** and **CC-T2**, the *Interactive* feature helps us to focus on two or more observable components without being distracted by the other observable components. For example, we focus on pc, have_privilege, and requesting of nodes by using the **focus** function as shown in Fig. 5. Based on the tip **CC-T1**, we use the **hide** function to focus on one of the three observable components. By observing graphical animations, we conjecture some characteristics as follows:

> Characteristic 1: There is at most one node that is located at cs, l6, l7, l8, or l9.
>
> Characteristic 2.1: There exists a case such that three nodes do not own the privilege.
>
> Characteristic 2.2: If a node owns the privilege, there is no other nodes that owns the privilege.

Based on **CC-T2**, we focus on two of the three observable components shown in the Fig. 5. By observing graphical animations, some characteristics are conjectured as follows:

> Characteristic 3.1: If a node is located at cs, l6, l7, l8, or l9, then the node owns the privilege.
>
> Characteristic 3.2: If a node is located at l3, l4, or l5, the node does not own the privilege.
>
> Characteristic 4.1: If requesting of a node is false, the node is located at rem or l1.
>
> Characteristic 4.2: if a node is located at rem or l1, requesting of the node is false.

In the search phase, the *pattern matching* feature is a candidate for **CC-T3**. We use this feature to find states in which there exists a privilege message in the network. The following figure shows a command that is used for this case.



There are 2 sequences

pattern:

(nw: NW:Network)
OCs:Config

condition:

hasPrivilege(NW:Networ
k)

where the top of the figure shows information of the feature, such as the **Pattern matching on all sequences** function and a number of sequences on a list. A command in the top rectangle and a command in the bottom rectangle are the pattern and the conditions, respectively. In the bottom rectangle, hasPrivilege in the command hasPrivilege(NW:Network) represents a function that checks whether the privilege is in the network written in the specification. Using **CC-T2** and observing the animations of such states shown in Fig. 6, we conjecture some characteristics as follows:

> Characteristic 5: There is only one privilege message in the network.
>
> Characteristic 6.1: If there is a privilege message in the network, no node owns the privilege.
>
> Characteristic 6.2: If a node owns the privilege, there is no privilege message in the network.
>
> Characteristic 7: If there is a privilege message in the network, no node is located at cs, l6, l7, l8, and l9.
>
> Characteristic 8: If there is a privilege message in the network, requesting of a receiver of the privilege message is true.
>
> Characteristic 9: If there is a privilege message in the network, a receiver of the privilege message is located at l4 or l5.

Note that we can conjecture the characteristics 6.1 and 6.2 by observing states satisfying the characteristic 2.1. The characteristic 7 can be guessed by the characteristics 6.1 and 3.1. To this end, by using the *Special display* feature for queue, ln, and rn observable components, we can conjecture some characteristics that are relevant to those observable components. First, using **CC-T3**, we use the *Pattern matching* feature to search states where queue is not equal to empty. Using **CC-T2**, and observing animations of the output, such as in Fig. 6 we can conjecture some characteristics as follows:

> Characteristic 10: Assume that I and J are a node that owns the privilege and a node that is the top of the queue of node I, respectively. The element at index J of the array ln of node I is less than the element at index J of the array rn of node J by one. The following figure shows an example of the characteristic where node J is N1 and index J is 1.

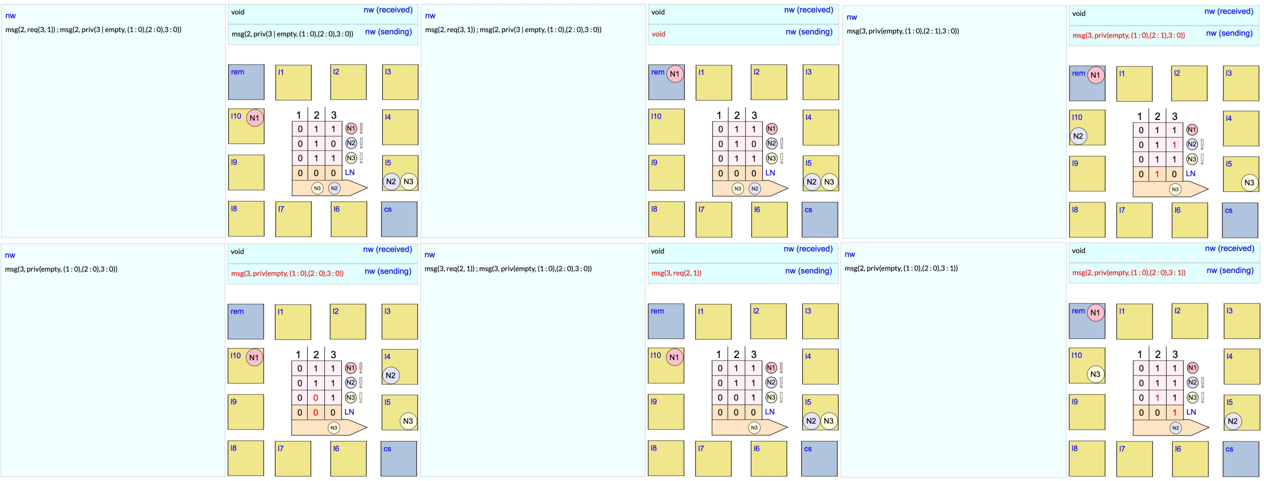
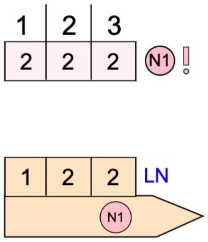


Figure 6. Some state pictures are returned from *pattern matching* feature



Characteristic 11: Each element in a queue of a node that owns the privilege is located at `l4` or `l5`.

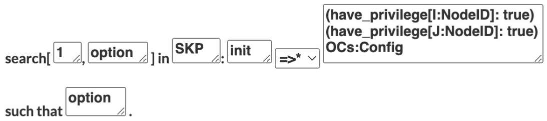
Note that we just list some of characteristics as examples for new and revised features to show the usefulness of those features.

C. Confirmation of Guessed Characteristics Based on the New Features

In the previous version, users can confirm guessed characteristics by using the Maude search command. In r-SMGA, users can also confirm the guessed characteristics by using the function **Search command**. For example, the command to confirm the characteristic 2.1 is as follows:

```
search [1] in SKP : init =>* (have_privilege[I]: true)
(have_privilege[J]: true) OCs:Config .
```

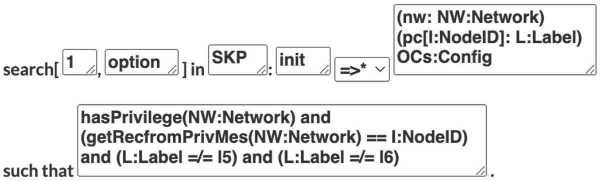
where `SKP` is a module, `init` is an initial state, `I` and `J` belong to sort NodeID, and `OCs:Config` represents other observable components. The command can be used in **Search command** of r-SMGA as follows:



search[1 , option] in SKP : init =>* (have_privilege[I:NodeID]: true) (have_privilege[J:NodeID]: true) OCs:Config

such that option .

The function tries to find a state in the state space in which two privileges occur. The function does not return any counterexample and hence the guessed characteristic is confirmed. The following figure shows the command for confirming the characteristic 9:



search[1 , option] in SKP : init =>* (nw: NW:Network) (pc[I:NodeID]: L:Label) OCs:Config

such that hasPrivilege(NW:Network) and (getRecfromPrivMes(NW:Network) == I:NodeID) and (L:Label =/= l5) and (L:Label =/= l6) .

where `hasPrivilege` and `getRecfromPrivMes` are two operations in the specification to check whether the network contains a privilege message and to get a receiver from the privilege message, respectively. For the other guessed characteristics, we also confirm them by using **Search command**. Note that users need to check guessed characteristics because they may not be correct. For example, we have conjectured that if one node sends the request message, this node will receive the privilege message from other nodes. We use **Model checking** to confirm the characteristic by following formula:

```
((reqMsgInNw(1)  |-> privMsgInNw(1))) /\
((reqMsgInNw(2)  |-> privMsgInNw(2))) /\
((reqMsgInNw(3)  |-> privMsgInNw(3))) .
```

where `reqMsgInNw(I)` is a proposition denoting that if the network contains the request message sent by node `I`, the proposition is true; otherwise, it is false, `privMsgInNw(I)` is a proposition denoting that if the network contains the privilege message sent to node `I`, the proposition is true; otherwise it is false, `_|->_` and `_/\_` are Maude operators that express leads-to and conjunction, respectively. **Model checking** returns a counterexample showing that the characteristic is not correct. When observing the counterexample by animations, we can clearly understand the situation. Therefore, r-SMGA can be considered as a visualization tool for understanding a counterexample better.

A video introduces r-SMGA and shows the usefulness of r-SMGA in conjecturing characteristics, which can be seen at: https://www.youtube.com/watch?v=MvyG6nmpOXs

VI. Related Work

Some previous work [4], [5], [6] have indirectly shown the potential of interaction in conjecture characteristics via SMGA. First, Bui and Ogata have introduced the Find Patterns feature [4]. Then, they have used the feature with their tips to conjecture non-trivial characteristics [5]. Moreover, interacting with elements in SMGA has been also mentioned in the work [6]. Particularly, it is difficult to visualize some information at the same time in SMGA, such as conflict and concurrent lanes in the autonomous vehicle protocol [10]. Therefore, the authors in the work [6] have shown that users can get those information by clicking on some elements. While observing animations and clicking on some elements, they have conjectured some non-trivial characteristics.

Tree graphs are a common kind of diagram to visualize some information that link together. A. Hernando et al. [11] have proposed a novel method using a tree graph to visualize huge information from related documents, such as news. Keywords or sentences are nodes where each node can be shown as raw texts or related images. There is one main node that is displayed as a picture containing related texts and images. The other nodes are displayed as raw texts or displayed same as the main node in which the main node is displayed more largely than the others. Users can observe nodes and navigate the graph to understand the relations of such nodes. When users navigate the graph, the main node is updated to let users easier to observe. ABETS [12] is a prototype for checking the correctness of Maude programs. The main purpose of the work is to improve the diagnosis of erroneous Maude programs. It uses tree graphs to visualize state sequences when nodes and edges correspond to states and rules, respectively. If an error occurs, the tool generates a tree graph that contain states which lead to the error. To understand the error, users can observe paths and click on states to expand the information of such states displayed as raw texts. One direction of our future work is to combine both approaches above to r-SMGA where state sequences can be used similarly to the work [12], be displayed similarly to the work [11], and be graphically animated by our approach when users select one concrete state sequence.

VII. Conclusion

We have integrated SMGA and Maude so that the revise version of SMGA (r-SMGA) can use some powerful features of Maude to support in conjecturing characteristics. By using Maude, the pattern matching feature of SMGA is revised to be able to handle associative-commutative pattern matching, such as searching some specific messages in the network, which cannot be dealt with regular expressions in the previous version. Some more interactive features have been provided to help users to concentrate on some elements in which users are interested. By graphically animating the Suzuki-Kasami distributed mutual exclusion protocol, we have demonstrated the usefulness of new and revised features in conjecture characteristics using the tips in [5] as guidelines. It is necessary to have a methodology to use the r-SMGA in reality and so one future direction is to find such methodology. Another future direction is to use the characteristics found by r-SMGA to prove that the Suzuki-Kasami protocol enjoys the mutual property using CafeoOBJ [13].

References

[1] T. T. T. Nguyen and K. Ogata, “Graphical animations of state machines,” in *15th DASC*, 2017, pp. 604–611.

[2] K. W. Brodlie, et al., Ed., *Scientific Visualization: Techniques and Applications*. Springer, 1992.

[3] E. Dimara and C. Perin, “What is interaction for data visualization?” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 119–129, 2020.

[4] D. D. Bui and K. Ogata, “Graphical animations of the Suzuki-Kasami distributed mutual exclusion protocol,” *JVLC*, vol. 2019, no. 2, pp. 105–115, 2019.

[5] ——, “Better state pictures facilitating state machine characteristic conjecture,” *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 237–272, 2022.

[6] D. D. Bui, et al., “Graphical animations of the Lim-Jeong-Park-Lee autonomous vehicle intersection control protocol,” *JVLC*, vol. 2022, no. 1, pp. 1–15, 2022.

[7] M. Clavel, et al., Ed., *All About Maude*, ser. LNCS. Springer, 2007, vol. 4350.

[8] I. Suzuki and T. Kasami, “A distributed mutual exclusion algorithm,” *ACM TOCS*, vol. 3, no. 4, p. 344–349, 1985.

[9] R. Rubio, “Maude as a library: an efficient all-purpose programming interface,” in *14th WRLA*, 2022, to appear.

[10] J. Lim, et at., “An efficient distributed mutual exclusion algorithm for intersection traffic control,” *J. Supercomput.*, vol. 74, no. 3, pp. 1090–1107, 2018.

[11] A. Hernando, et al., “Method to interactively visualize and navigate related information,” *Expert Systems with Applications*, vol. 111, pp. 61–75, 2018.

[12] M. Alpuente, et al., “Debugging Maude programs via runtime assertion checking and trace slicing,” *J. Log. Algebraic Methods Program.*, vol. 85, no. 5, pp. 707–736, 2016.

[13] R. Diaconescu and K. Futatsugi, *CafeOBJ Report*. World Scientific, 1998.

An Accurate and Robust Step Detection Method Based on Continuous Wavelet Transform

XiangChen Wu, Yang Zou*, Xiaoqin Zeng, Xiaoxiang Lu, Keman Zhang
Institute of Intelligence Science and Technology, School of Computer and Information,
Hohai University, Nanjing, China
{wxc, yzou, xzeng, luxx0824}@hhu.edu.cn

Abstract—**The existing step detection algorithms have rarely been designed for walking in complex scenes. This paper presents a precise method for detecting steps wcouldith multiple walking gaits in scenes that involves a variety of terrains. The method consists of three phases. First, the Kalman filter is adopted to denoise the raw data. Then, a continuous wavelet transform is applied to the filtered data to attain the obvious gait pattern in the time spectrum. Finally, a robust detection algorithm is proposed to implement step counting and single stride segmentation. The dataset is established from the experiments where 14 adults of diverse heights and weights continuously walked with multiple gaits on a variety of terrains. The raw data is generated by the Inertial Measurement Unit bonded on participants' calves. The experimental results show that the average accuracies of step counting achieved by our method for level-walking, up and downstairs, and mixed complex situations are 99.0%, 98.2%, and 99.1%, respectively, and effective results of single stride segmentation are also suggested.**

Keywords-step detection; wavelet transform; pedestrian dead reckoning; inertial measurement unit

I. INTRODUCTION

As the prerequisite of step analysis, the step detection process in wearable devices has always been a research focus, which means accurate step cycle segmentation and step counting will serve step analysis well [1]. Step detection is also an important requirement in many other fields, and accurate step counting in most scenes of daily life is an indispensable demand to ensure the accuracy of some systems. In indoor positioning [2], step counting is an essential element for Pedestrian Dead Reckoning (PDR). In the medical field, the result of step counting is utilized as a measure of the rehabilitation situation of patients, and accurate step counting results obtained from any walking paths would help further optimize the rehabilitation plans for patients [3]. Furthermore, the result of step counting is often used as a reference standard for the training arrangement of athletes [4]. However, accuracy step counting has always been a challenge as it is not independent of changes in pedestrians' gaits in walking and scenes in which they walk.

At present, most step counting algorithms are designed based on thresholds [5-7]. In this way, it is impossible to take gait changes and walking scenes into account, as doing so may result in possible structural step losses [8]. Therefore, this kind of method cannot be directly generalized to tasks of different people walking in complex scenes. In addition, these algorithms often require a time-consuming process of dynamic threshold setting [9]. Threshold detection is a classical implementation of the gait detection method based on standing detection, which is carried out to detect the moment when a foot is placed on the ground during walking [10]. It could be influenced by the additional factors associated with the deformation and elasticity of the shoes being worn, making it difficult to conduct an appropriate threshold setting. In the experimental stage, the accuracy of the algorithms is mostly derived in terms of a specified number of steps in a laboratory scenario. Hence, these methods cannot fully capture step features of ordinary people's daily walking [11-12].

To overcome the above-mentioned shortcomings, a robust step detection method is presented in the paper to tackle ordinary people's steps with multiple gaits in complex scenes. First, since the sampled signal contains random noise, we use the Kalman filter to estimate a more realistic current state by using an equation concerning the previously measured value and a current state value. Then, through a comprehensive analysis and comparison of experimental results, a specific wavelet basis function is chosen, which can attain a robust stride representation in the frequency domain. Then the level-walking is taken as an example to analyze the data of gaits and representative characteristics of a single gait are obtained accordingly. Finally, based on the frequency-domain characteristics of wavelet transform with respect to the specific wavelet basis function, an efficient and accurate step detection algorithm is proposed.

In our experiments, we consider two categories of scenes that people may encounter during daily life, one is an office building and the other is a hill that includes an incline with various slopes, undulating grasslands with stairs, gravel pavements, and straight asphalt pavements. Participants of the experiments would walk in a variety of gaits on the preset paths including alternating multiple terrains. Experimental results show that the proposed algorithm can achieve accurate step counting results and suggest reliable outcomes for single stride segmentation in both ordinary and complex scenes.

The technical contributions of this paper are as follows:

- A general selection criteria of wavelet basis functions is presented to enhance the spectral cluster features of the cohesiveness and independence of a single step signal, experimentally.
- A fast and accurate gait detection algorithm is proposed based on the spectral cluster features with $O(n)$ time/space complexity.

*Corresponding author: yzou@hhu.edu.cn (Y. Zou)

DOI reference number:10.18293/DMSVIVA22-013

- A walking dataset is constructed that contains complex terrains and it is verified through experiments that complex terrains can bring about more complicated state changes in walking and consequently cause detection algorithms less effective.
- The experimental results show that the average accuracies of step counting achieved by our method for level-walking, up and downstairs, and mixed complex situations are 99.0%, 98.2%, and 99.1%, respectively, and an effective single stride segmentation is also suggested.

The remainder of this paper is organized as follows. Section 2 presents the system architecture, including the hardware, placement of sensors, and data collection; Section 3 proposes the algorithm for step detection; Section 4 conducts the experiments; Section 5 analyzes the results; and finally Section 6 concludes the paper.

II. System Architecture

A. System Overview

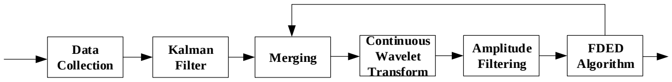
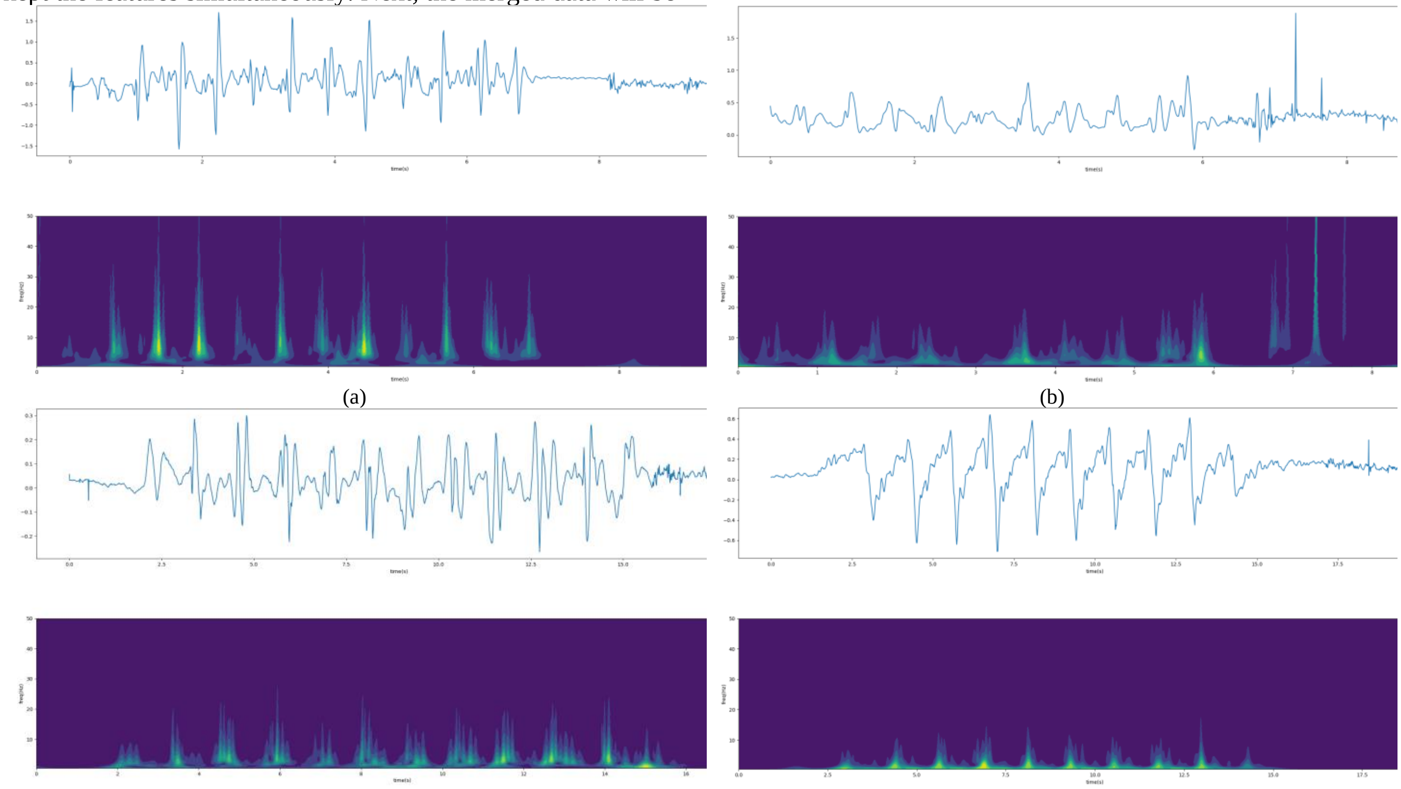


Figure 1. The architecture of the step detection method.

The architecture of the proposed step detection method is displayed in Fig. 1 The raw data is collected from the sensor mounted on the participants' calves. It will be put into the Kalman Filter and then be merged to filter out the noises and kept the features simultaneously. Next, the merged data will be transformed from the time domain into the frequency-domain by a continuous wavelet. In the frequency domain, an adaptive amplitude filtering procedure is designed dynamically which could separate the cluster that represents a continuous stride. Finally, a step detection algorithm is proposed based on the human walking transition features in the frequency domain that could count precisely no matter the complex walking environment or mixed gait processing.

B. Hardware

We use the WT901SD board created by WIT-MOTION as the basic hardware of the sensor. It involves a motion processing unit, a data storage unit, and the main control unit. The motion processing unit encapsulates a 3-axis accelerometer, a 3-axis angular velocity meter, a 3-axis magnetic field, and a 3-dimensional angle on a cuboid with a length, width, and height of 27mm, 19.5mm, and 7.2mm respectively. This board utilizes an SD card for long-term storage and requires a battery with a voltage of 3.3V~5V, and a current <40mA to keep the board running for a long time. The main control module of the board is e230, which is responsible for providing control requirements during sensor sampling. The range of the accelerometer is ±2g, ±4g, ±8g, ±16g (optional), the angular velocity of the gyroscope is ±250/500/1000/2000°/s (optional), and the sampling frequency of each component of the sensor can be selected in the range of 0.1Hz~200Hz. In the process of our experiments, we compared and analyzed the influence of sampling frequencies and found that maintaining a larger sampling frequency would be more convenient to implement our algorithm.



(a)

(b)

(c)

(d)

Figure 3. The images (a)-(d) are the characteristics in the frequency domain that corresponds to the situations in which the seniors are placed on the thigh, left- arm, waist, and calf, respectively.

C. *Sensor Placement and Data Collection*

To keep the balance between the participants' comfort in walking and the representativeness of the collected data, we chose to place the sensor on the calf being close to the knee, as shown in Fig. 2 only one sensor is used throughout the experiments. The complete walking processes of all the participants are filmed by video equipment for later comparison between the actual walking steps and the detection results.

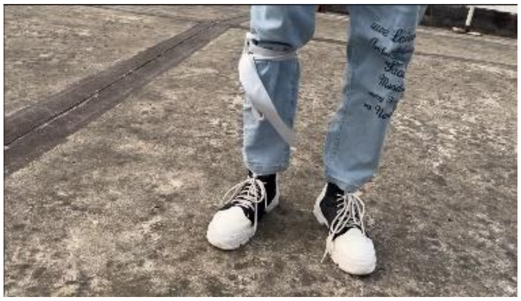


Figure 2. the placement that the sensor bonded on the calf near the knee.

The step detection method proposed only relies on the output of the 3-axis accelerometer, and the sampling data is temporarily stored in the SD card. The board is placed horizontally, the sampling frequency is set to 200Hz, and the corresponding bandwidth is set to 188hz. Throughout the experiments, all the participants were required to walk freely with the sensor bonded on their right calf near the knee. Although the sensor was placed on the calf in the experiments, we verified that accurate counting results could also be attained by the proposed algorithm when it was placed on the human torso or lower limbs. However, through comparison of experimental results, we found that the more obvious the characteristics of collected walking signals in the frequency domain are, the more burdened the sensor placed on the human muscle group during the walking process is. Fig. 3 shows the characteristics in the frequency domain corresponding to the four parts: thigh, left arm, waist, and calf, on which the sensor is worn in horizontal mode. It can be seen that the characteristics in the frequency domain of the pedometer signals are mixed when the sensor is worn on the left arm, whereas they are clearer when the sensor is placed on other parts.

III. METHODOLOGY

The method for step detection is composed of three phases. First, the wavelet basis function is determined in terms of the characteristics of independence and completeness in the frequency domain of gaits. Then, the sampled signals are preprocessed to enhance the feature patterns of strides. Finally, a gait detection algorithm is proposed to implement step counting and suggest stride segmentation according to these patterns.

A. *Continuous Wavelet Transform*

Suppose the Fourier transformation of ψ(t) is ψ(ω) satisfying the permissibility relation, then we call ψ(t) as the wavelet basis function or basic wavelet function. Most wavelet basis functions could be defined by (1), where φ is a predefined scaling function that constitutes the integer translation orthogonal basis of a wavelet function, and hψ(k) is the wavelet coefficients which are an ordered set.

$$\psi(x) = \sum_k h_\psi(k)\sqrt{2}\varphi(2x-k) \tag{1}$$

Wavelet transforms defined in (2) allow us to transform the analysis of one-dimensional signals into the processing of pole numbers. The wavelet transformation procedure is expressed in the form of dimension increase, but its analysis relies on the specific finite numbers that simplify the process. Under the condition of reversibility of wavelet transforms, analysis of the spectrum after wavelet transform will be equivalent to the processing of the original signal. However, the self-adaptive observable window’s width of wavelet transform also brings about the characteristic of multi-resolution, which causes the low-time resolution and high-frequency resolution of slow-changing signals and the high-time resolution and low-frequency resolution of rapidly changing signals [13].

$$\mathrm{W}_f(a,b) = \int_{-\infty}^{+\infty} f(t)\overline{\psi}_{(a,b)}(\mathrm{t})\, dt \tag{2}$$

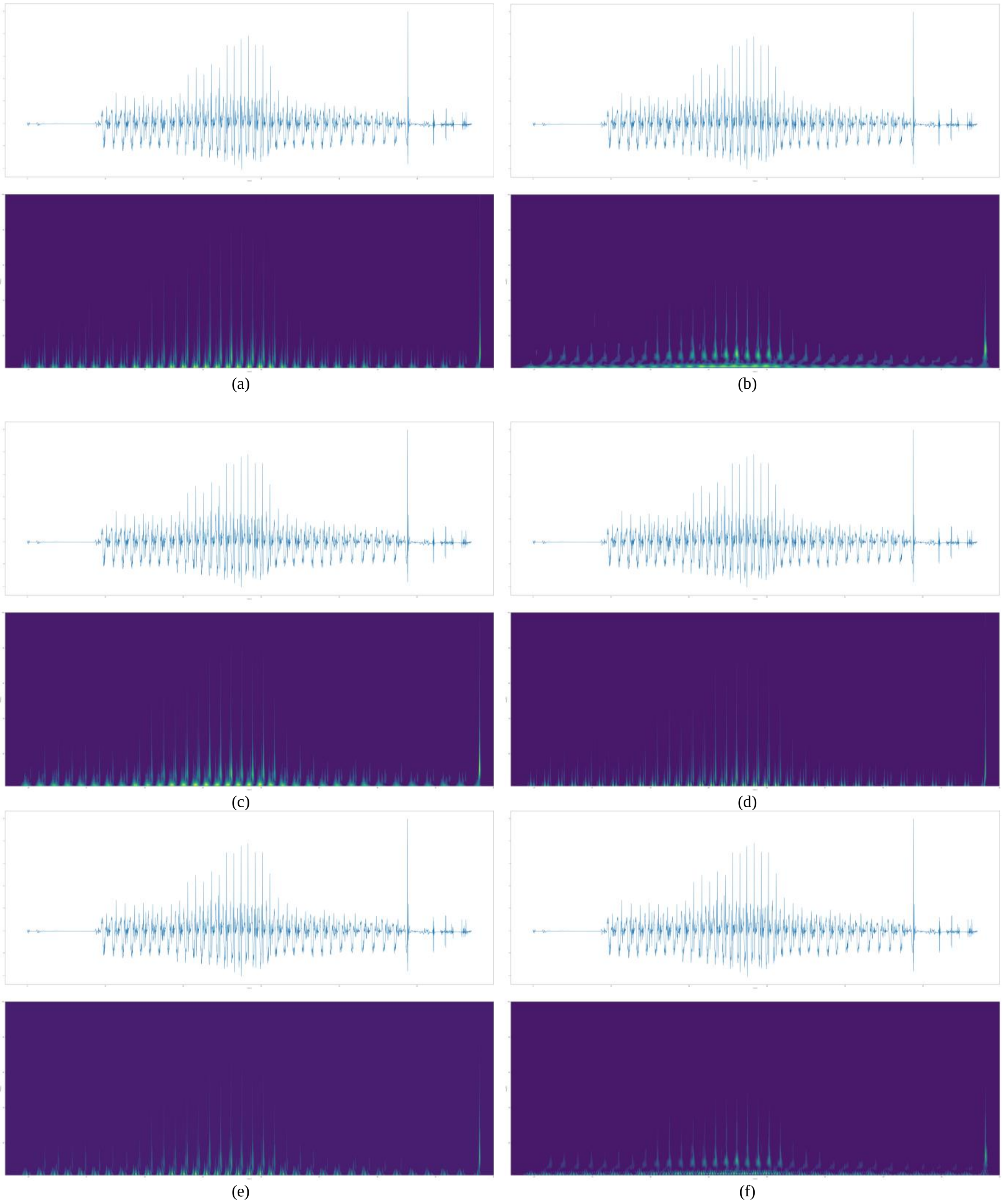
With the development of wavelet function, there are many choices of wavelet basis function. Here we choose the wavelet basis function in terms of the clustering effects of stride signals in the frequency domain. Fig. 4 (a)-(f) show the images of the frequency domain after the wavelet transforms with ‘cgauP’, ‘cmor’, ‘fbsp’, ‘gausP’, ‘mexh’, ‘morl’ as basic wavelet functions respectively, were made to a certain stride’s signals. We took the characteristics of independence and completeness of the clustering phenomenon (cluster) of strides’ signals in the frequency domain as the basis for the selection of the wavelet basis function. The former characteristic indicates that signals in the frequency domain of adjacent strides are less correlated, whereas the latter indicates that signals in the frequency domain representing one stride are sufficient to distinguish. By analyzing the experimental results, we chose cgau1 as the continuous wavelet basis function. With the redundancy caused by a continuous sampling of scale parameters by the continuous wavelet transform in (2), the spectrogram of the signals would be more readable. The formula of cgau1 is expressed as (3), where 1 means to take one derivative of this function.

$$\psi(t) = c\left(-ie^{-t^2-it} - 2te^{-t^2-it}\right) \tag{3}$$

B. *Signal preprocessing*

As shown in Fig. 1, before step detection, a series of signal preprocessing needs to be carried out on the raw data to obtain more effective frequency domain features.

1) Kalman filter: The signal data sampled in each time slice is filtered by the Kalman filter to obtain the relative truth value for the preliminary processing of random noise. Kalman filter maintains a set of state equations and measurement equations to estimate a more real current state according to the past measured values and the current state values. This task will be implemented directly in the hardware system.


(a)

(b)

(c)

(d)

(e)

(f)

Figure 4. The images of the frequency domain of a stride’s signals after wavelet transforms are made with basic wavelet functions ‘cgauP’, ‘cmor’, ‘fbsp’, ‘gausP’, ‘mexh’, and ‘morl’, respectively.

2) Continuous Wavelet Transform: To ensure that the walking signal has a higher energy aggregation state on the spectrogram and facilitate the design of the next algorithm, we select Cgau1 as the wavelet basis function. With the aid of the reversibility feature and redundant representation of continuous wavelet transform, we directly perform the step detection process in the frequency domain. The feature of multiresolution of continuous wavelet transform can be utilized to design the step detection algorithm in the non-threshold form.

3) Amplitude Filtering: Considering that the proposed algorithm for step detection relies on an independent clustering structure of a stride, we perform spectral value filtering on spectral information for each time slice. To ensure the independence of each stride's signals, we need to set the lower spectral values of a small part of the data to 0 in combination with the spectral value distribution of the current time slice, without affecting the overall clustering structure. Most of the information with lower spectral values appears in gaps of strides' signals due to random noise, and that with high spectral values are also affected to some extent.

C. Algorithm

The step detection process is mainly completed in the frequency domain and relies on the coefficient representation of continuous wavelet transform. According to the upper-frequency limit of the walking process, we set the value of parameter totalscale, which determines the continuous scale parameter change, to 256. The step detection algorithm proposed is named as frequency domain extension detection algorithm (FDED), and the process of step counting does not depend on the threshold setting.

```
Algorithm 1 FDED
Input: Time spectrum $D_i$, the remaining of last time $r_{i-1}$, $i > 1$;
Output: Partition set
  Preset $\epsilon, \beta$;
  Initialize array $P1, P2, P3$;
  Perform $AmplitudeFiltering$ on $D_i$ to get the filtered data $d_i$;
  Put $d_i$ into $P1$;
  while $P1$ is not null do
    Take out element $p$ from $P1$;
    if $Low\_Scale\_Support(p) > \epsilon$ then
      Put $p$ into $P2$;
    end if
  end while
  Perform $Merging$ on $P2$ and $r_{i-1}$ ;
  while $P2$ is not null do
    Take out element $q$ from $P2$;
    if $Multiline\_Extension(q) > \beta$ then
      Put $q$ into $P3$;
    end if
  end while
  Perform $Segmentation$ on $P3$ to get the remaining $r_i$ and results;
  Put results into partition set;
```

The main purpose of the algorithm is to obtain temporal region proposals for the frequency domain "cluster" generated by the walking process. From the reversibility of the wavelet transform, the regional proposals of the "cluster" in the frequency domain can be exploited as the recommended walking time interval for a stride. Algorithm FDED for step detection is shown above and the procedures involved are elaborated as follows:

- Select all the time points with valid high spectrum values exceeding the scale parameter value ε in the current time slice to form a set P1. At this point, P1 mainly contains the third part of the "cluster" of walking steps and random high-frequency noise.
- Preprocess the set P1, and filter, merge and mean the adjacent time points.
- Perform low-frequency threshold judgment on all suggested points in P1 at this time to reduce random high-frequency noise, which is mainly based on the third part of the "cluster" of walking steps. The filtered set is named P2.
- Carry out Algorithm FDED to all the proposed points in P2 at this time and add the areas that meet the extension conditions to the area set P3.
- Match all suggested points in the set P3 to the threshold β for comparison, where the threshold β indicates the recommended period under normal walking conditions. And the time domain signal after the last complete region recommendation time is returned to the merging part for slice detection for the next time.

The two thresholds proposed in the algorithm, ε, and β, are set only to exclude impossible points, which mainly include the alternative points without low-frequency support and too small a segmentation period suggested that often comes from the truncated edge mapping of continuous wavelet transform.

Note that the procedure *Multiline_Extension* in Algorithm 1 can execute parallelly. The time and space complexities of Algorithm 1 are both $O(n)$ where n relies on the sampling frequency of the sensor, even if not performing in parallel mode.

IV. EXPERIMENT

In light of the category of scenes, the experiment dataset can be divided into two parts, one is used to investigate different walking states in an ordinary environment, called Env1; the other is used to investigate step detection under the condition of mixed walking in a complex environment, called Env2. Env1 is set in a general building, which is composed of cement floors and stairs. It is suitable for detecting the participants' level-walking and going up and downstairs in a relatively short distance. Env2 is set on a hill Diecui on the campus. As can be seen from the marks in Fig. 5, the sightseeing routes of Diecui include a variety of paths, such as inclined planes with various slopes, stairs, and rolling lawn with gullies, gravel pavement, and straight asphalt pavement, which is suitable for long-distance free walking, as required by the experiments. All the data constituting the dataset come from the sensor, and sampling signals are generated from the walking processes of the participants in these two kinds of experimental environments.

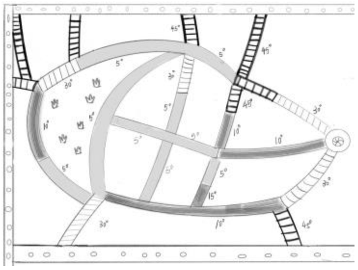


Figure 5. Top view of Diecui hill.

As can be seen from Fig. 5, the terrains comprising env2 are relatively complex, and seven experimental routes are designed according to the distribution of terrains and existing sightseeing, as shown in Fig. 6. Aiming to investigate the step detection tasks in different combinations of terrains, each of the routes is devised to cover 3 different terrains and the combination of terrains in each route is shown in Table I:

Table I. The combination of terrains for each route.

Items	Terrain elements of each route
1	stairs of 30°/45°, a slope of 5°, flagstone of 30°
2	stairs of 30°/45°, a slope of 5°, gravel road of 10°, flagstone of 30°
3	stairs of 10°/45°, flagstone of 30°, a slope of 5°
4	stairs of 65°/30°, flagstone of 10°/30°, a slope of 5°
5	stairs of 30°/45°, lawn, a slope of 5°, flagstone of 10°
6	stairs of 30°/45°, flagstone of 10°/15°, gravel road of 5°
7	stairs of 30°/45°, flagstone of 5°/10°/15°, a slope of 5°, gravel road of 5°

Table II. Participants' characteristics

Participant	Height(cm)	Mass(kg)	Gender
1	171	55	female
2	168	70	male
3	158	50	female
4	180	80	male
5	175	90	male
6	176	70	male
7	177	60	male
8	161	43	female
9	180	64	male
10	178	66	male
11	178	66	male
12	173	60	female
13	176	72	male
14	182	75	male
Mean	173.7	65.7	
Standard Derivation	7.1	12.1	

Participants are 14 adults, including 10 men and 4 women. As shown in Table II, their heights range from 158cm to 182cm and their weights range from 40kg to 80kg, and the mean and standard deviation of their heights and weights are 173.7cm, 7.1cm, 65.7kg, and 12.1kg, respectively. The dataset is composed of the data from 14 individuals walking with diverse gaits on various routes, including going up or downstairs, level-walking, and non-level walking, jumping, hopping, and various mixed walking procedures.

In experiments, it is not appropriate to require participants to walk in a constant number of steps, because high-frequency dimming will always appear in the corresponding spectrum obtained in this way, and low-frequency aggregation will decline when it finally approaches the specified number of steps. The occurrence of this phenomenon is easy to understand. As the participants are responsible for counting steps, when they approach the specified number of steps, they always deliberately reduce the pace and take a more cautious way of walking. Therefore, our experiments are designed based on free walking, which aims at the predefined destination, and the participants can choose to stop walking when they are near the destination. This is more in line with real situations in normal walking. we record all the step videos, which could be used to not only analyze the walking process but also to provide accurate values of step counting.

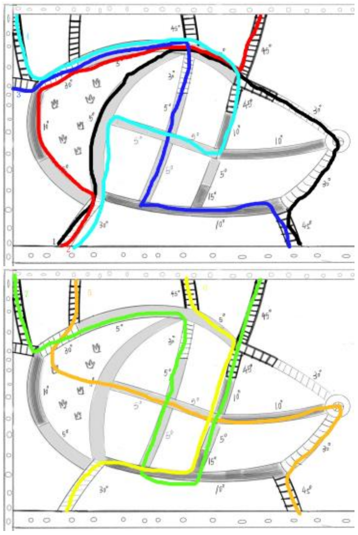


Figure 6. There are seven walking routes designed on Diecui, and each route is a mix of a variety of terrains.

V. Experimental Results and Analysis

According to the category of scenes, the experimental tasks are partitioned into two sets. One is the step counting task in a general building. This stage mainly examines the relatively single walking status, such as the step counting and step cycle segmentation effects of level-walking, going upstairs and downstairs. In this scene, the change of gaits performed by the participants is slow walking, regular walking, fast walking, and variable-speed mixed walking. Since variable-speed mixed walking is a challenge for most pedometer algorithms, especially when the participant's switching from fast walking to slow walking frequently causes missed counts. So, the walking task with variable speed is treated as a column of level-walking steps. To ensure that the experimental data can generally reflect the effectiveness of the proposed step detection algorithm, each participant in Env1 was required to complete the same walking state three times under the same conditions. The accuracies of step counting and suggestion of stride segmentation are all taken on average.

Table III gives the experimental results of all participants in Env1, and all the results keep two decimal places. We use two classical evaluation metrics for algorithm validation. One is step counting accuracy (Ac), which is the ratio of the number of steps counted by the algorithm to the true walking steps. The other is the scale in single stride segmentation (Pc), which is the ratio of the period calculated by the algorithm to the true period of a stride. The calculation formulas are defined as follows: where $\underline{N_{true}}$ is the real number of steps, N_{count} is the number of steps detected by the algorithm, P_{true} is the real step duration, and P_{seg} is the suggested period for the stride segmentation given by the algorithm.

$$\mathrm{Ac} = \left(1 - \frac{\left|N_{\mathrm{true}} - N_{\mathrm{cout}}\right|}{N_{\mathrm{true}}}\right) \times 100\% \qquad (4)$$

$$\mathrm{Pc} = \left(1 - \frac{\left|P_{\mathrm{true}} - P_{\mathrm{seg}}\right|}{P_{\mathrm{true}}}\right) \times 100\% \qquad (5)$$

Table III. The experiment results in Env1 where the level-walking is divided into four categories, slow speed, regular speed, fast speed, and variable speed, and stair walking is divided into two categories, ascending and descending.

Participant	Level walking								Stair walking			
	slow speed		regular speed		fast speed		variable speed		ascending		descending	
	Ac	Pc	Ac	Pc	Ac	Pc	Ac	Pc	Ac	Pc	Ac	Pc
1	99	95	99	92	99	82	99	84	99	95	96	88
2	98	92	99	95	100	74	99	90	100	92	97	90
3	100	96	99	92	99	81	98	83	98	87	98	91
4	100	93	98	93	99	75	98	79	100	94	97	86
5	99	96	100	92	100	76	99	91	98	89	95	91
6	99	95	100	94	98	81	99	86	97	96	97	92
7	100	91	99	88	99	79	99	84	100	91	98	88
8	100	92	99	94	99	82	98	92	99	93	100	90
9	98	95	100	92	98	77	99	84	98	88	98	88
10	99	87	99	84	99	83	99	78	99	95	97	87
11	100	92	98	87	100	79	98	87	100	94	100	88
12	99	95	99	90	99	72	100	86	100	96	98	86
13	99	94	99	91	98	78	99	85	98	89	97	85
14	99	92	99	88	99	82	98	82	99	93	97	87
Average	99.2	93.2	99	90.9	99	78.6	98.5	85.1	98.9	92.2	97.5	88.2
Maximal error	2	13	2	16	1	28	1	22	3	13	5	15

Table IV. Experimental results in complex environment Env2 where the second column contains all the routes that each participant chooses.

Participant	Routes	AM	AD	FDED
1	1,2,6	94.4	82.7	99.6
2	3,4,7	93.1	84.1	99.5
3	4,5,6	98.2	94.1	99.1
4	1,2,7	98.9	83.3	99.4
5	1,5,6	97.5	90.1	98.6
6	2,4,6	95.9	83.8	99.2
7	3,5,7	97.4	86.6	99.4
8	1,3,7	98.4	86.5	98.4
9	2,4,6	91.5	91.2	99.5
10	2,4,7	84.1	85.3	99.3
11	2,4,6	87.7	86.1	99.4
12	1,3,4	95.2	81.8	98.8
13	1,5,7	97.3	92.5	99.4
14	2,3,4	97.1	85.9	98.7
Average		94.7	86.7	99.1
Maximal error		15.9	18.2	1.6

From the figures of Average and Maximal error in the table, it can be seen that the average accuracies of level-walking at various single speeds are not much different from each other, and they are at a relatively high level of counting accuracy. However, from the proportion of suggested single stride segmentation, the average of Pc is lower than that of Ac, because during a normal walking cycle the frequency domain walking "cluster" can only be described by the forward movement of the center of gravity caused by walking with unbound legs. The counting accuracy of going upstairs and downstairs are 98.9% and 97.5% respectively, but the Pc are all lower than level-walking, which are 92.2% and 88.2% respectively. This is mainly because, in the process of going upstairs and downstairs, the alternating motion frequency of legs is higher than that of normal level-walking, which corresponds to a less complete representation of a stride described by the walking "cluster" in the frequency domain. This phenomenon happens at a fast speed of level-walking as well, because in those situations there is a contradiction between Ac and Pc. According to Table II, in the process of mixed variable-speed walking, we can see that the counting accuracy of variable-speed walking has no obvious difference with the changes of various factors of the participants, which shows that the representation of the walking feature is effective and the proposed algorithm is robust.

Table V. Experimental results in the complex environment Env2 according to the breakdown by route.

Route	AM	AD	FDED
1	96.4	84.1	98.9
2	92.5	84.1	99.4
3	96.4	84	99.1
4	92.1	87	98.7
5	98.9	92.5	99.5
6	94.5	91.1	99.5
7	95.3	83.3	99.6
Average	95.1	86.5	99.2
Maximal error	7.9	16.7	1.3

Table IV and Table V show the results of experiments separately completed by all participants in Env2. In Table IV, all the participants were asked to randomly select three of the seven paths and complete the walk in a free manner. The step counting results based on the breakdown by route are given in Table V. The complete walking process of each participant was recorded by the camera for later walking counting. Two typical algorithms will be used for comparison, one is a rule-based detection algorithm [9], abbreviated as AM, and the other is based on dynamic step jerk settings which constitutes a gait detection software on Android devices, abbreviated as AD. As can be seen from the table, under the switching of complex row paths, the AM algorithm fails to maintain good stability, and it’s step counting accuracy is poor in routes 2 and 4. This may be due to that the threshold-based gait detection algorithms often leads to structural gait loss. Specifically, the threshold setting methods based on the detection of a walking period frequently causes lost steps because the dynamic change of the threshold during a shifting process from slow to rapid walking cannot be responded in time. Moreover, since AD algorithm relies on step jerk, and Env2 contains a lot of slopes, the walker will behave more cautiously at the end of each step when the heel touches the ground, which is not conducive to the detecting process of AD.

However, in the proposed algorithm, the delay caused by indirect response of the unbound leg provides enough time for the FDED algorithm to accurately capture the true walking process, which is also reflected in the walking experiment with variable speeds. In addition, it is experimentally verified that tying the sensor to the participant's torso and thighs can produce stable frequency domain "clusters" while maintaining a certain sampling frequency and these features can also be recognized by the FDED algorithm as gaits.

Finally, even when applied to the situations of walking on combined routes with complex terrains, FDED achieves a considerably high accuracy of step counting, with an average accuracy of 99.1% and a maximum error of 1.6%, which is much higher than AM and AD. The detection speed of FDED is similar to AD and faster than AM in running time. Therefore, the proposed algorithm has strong adaptability in complex environments.

VI. CONCLUSION

This paper has proposed an efficient step detecting method that can detect walking with multiple gaits in complex scenes with a variety of terrains. First, through experiment comparison, a specific wavelet basis function is selected to map the walking signals sampled by the sensor to the frequency domain. Then, based on the feature patterns achieved by preprocessing in the frequency domain, an algorithm FDED is proposed to implement step counting and single stride segmentation. Experimental results have shown that FDED can attain considerably accurate step counting results and suggest effective single stride duration in both ordinary and complex scenes. Different from other detection methods, FDED does not depend on the threshold setting and thus can be effortlessly generalized to different people and scenes.

REFERENCES

[1] H. Zhao, Z. Wang, S. Qiu, J. Wang, F. Xu, Z. Wang, Y. Shen, “Adaptive gait detection based on foot-mounted inertial sensors and multi-sensor fusion,” Information Fusion, 2019, 52:157-166.

[2] W. Suksuganjana, S. Laitrakun, K. Athikulwongse, Y. Hara-Azumi, and S. Deepaisarn, “Improved Step Detection with Smartphone Handheld Mode Recognition,” 2021 13th International Conference on Knowledge and Smart Technology,” 2021, pp. 55-60.

[3] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, PJM. Havinga, “Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey,” In 23th International conference on architecture of computing systems 2010 (pp. 1-10), February ,2010.

[4] A. Tanigawa, S. Morino, T. Aoyama, M. Takahashi, “Gait analysis of pregnant patients with lumbopelvic pain using inertial sensor,” Gait & Posture, 2018, 65:176-181.

[5] J. Taborri, E. Palermo, S. Rossi, P. Cappa, “Gait partitioning methods: A systematic review,” Sensors, 16(1), 66.

[6] G. Fortino, S. Galzarano, R. Gravina, W. Li, “A framework for collaborative computing and multi-sensor data fusion in body sensor networks,” Information Fusion 22 (2015) 50–70.

[7] Z. Wang, D. Wu, R. Gravina, G. Fortino, Y. Jiang, T. Kai, “Kernel fusion based extreme learning machine for cross-location activity recognition,” Information Fusion, 37, 1-9.

[8] AR. Anwary, H. Yu, M. Vassallo, “Optimal Foot Location for Placing Wearable IMU Sensors and Automatic Feature Extraction for Gait Analysis,” IEEE Sensors Journal, 2018:2555-2567.

[9] V. Genovese, A. Mannini, AM. Sabatini, “A Smartwatch Step Counter for Slow and Intermittent Ambulation,” IEEE Access, 2017, 5:13028-13037.

[10] R. Harle, A survey of indoor inertial positioning systems for pedestrians[J]. IEEE Communications Surveys & Tutorials, 2013, 15(3): 1281-1293.

[11] W. Suksuganjana, S. Laitrakun, K. Athikulwongse, Y. Hara-Azumi, and S. Deepaisarn, “Improved Step Detection with Smartphone Handheld Mode Recognition,” 2021 13th International Conference on Knowledge and Smart Technology,” 2021, pp. 55-60.

[12] C.Soaz, Diepold K, “Step Detection and Parameterization for Gait Assessment Using a Single Waist-Worn Accelerometer,” IEEE Transactions on Biomedical Engineering, 2016, 63(5):933-942.

[13] S. Mallat, A wavelet tour of signal processing. Elsevier, 1999.

A method for detecting abnormal users with fake stars

Jing Jiang, Hao Li, Yifan Liu, and Li Zhang
The State Key Laboratory of Software Development Environment, Beihang University, Beijing, China

***Abstract*—In GitHub, users star interesting repositories, and the number of stars is viewed as the significant measure of repository popularity. Some repositories obtain fake stars by unjustified means, which ruin efforts that communities have made stars a valuable indicator, and bring negative impacts in GitHub. Therefore, it is important to stop abusing GitHub stars and detect abnormal users who provide fake stars. In this paper, we first define features from the user dimension and repository dimension. Then we perform differential analysis and find that most of the features show a significant difference between abnormal users and normal users. Next, we propose a method AUDetec for Abnormal User Detection. The method AUDetec uses the decision tree to detect the abnormal users based on two features, including the sum of repositories starred by the user and the median value of the number of days since creation for repositories starred by the user. We evaluate the effectiveness of AUDetec on the data set which contains 120 abnormal users and 240 normal users. The experiment results show that AUDetec has a high performance by achieving an accuracy of 99.86% on average.**

***Index Terms*—Abnormal user detection, Fake star, Open source software, GitHub, Repository popularity**

I. Introduction

GitHub is a famous social coding site for open source software and allows developers to follow interesting users to receive their activity updates. Besides relationships between developers, GitHub also supports relationships between developers and repositories. Inspired by the like button of modern social networks, developers star their interesting repositories for the purpose of showing appreciation to repositories, keeping track of updates, and discovering related context in news feed[1]. According to the previous work [1], three out of four developers consider the number of stars before using or contributing to repositories, and the number of stars is viewed as the significant measure of repository popularity in GitHub. Furthermore, some researchers select the GitHub repositories for their study base on the number of stars of repositories [2]. Therefore, the number of stars in a repository becomes a critical metric reflecting the repository's quality and popularity.

Due to the importance of stars, some repositories obtain fake stars by unjustified means. An open-source project is promoted via a free drink in return for a star[2]. Participants in this promotion get a chance of free Starbuck drink by starring this repository, which obtains nearly 2,500 stars. As the open-source community receives increasing recognition in companies, developers who have repositories with many stars are welcomed in the job market. A mutual star community GitStar emerges where developers star repositories for users who then provide fake stars in return. Some developers obtain a large number of fake stars in GitStar and successfully receive offers from big internet companies such as Alibaba[3]. Fake stars ruin efforts that communities have made stars a valuable indicator, and bring negative impacts in GitHub. Repositories with fake stars are overvalued, which affects the recognition of GitHub stars in the job market. Furthermore, fake stars may mislead developers into contributing to poor-quality repositories with fake stars. Therefore, it is important to stop abusing GitHub stars and detect abnormal users who provide fake stars.

There are literature about abnormal user detection [3], [4]. Thomas et al. detected abnormal users based on user behavior features [3]. However, these works are based on other platforms (e.g. Facebook) for other malicious activities, such as spam, phishing, or malware. Besides, Borges et al. understood repository starring practices in GitHub [1]. The number of stars is a vital metric for researchers to select popular repositories [2]. However, these works consider that repositories are starred by normal users, and ignore fake stars which are provided by abnormal users.

In this paper, we propose to detect abnormal users with fake stars in GitHub. According to previous work [1], we first define features from user dimension and repository dimension. Then we perform differential analysis and find that most of the features show a significant difference between abnormal users and normal users. Next, we propose an approach AUDetec for Abnormal User Detection. Based on experiment results, two features are enough to distinguish abnormal users and normal users, including starring_repository_number(the sum of repositories starred by the user) and repository_age (the median value of the number of weeks since creation for repositories starred by the user). Based on the above two features, AUDetec uses the decision tree to make classification and detect abnormal users.

In order to evaluate the effectiveness of AUDetec, we obtain 120 abnormal users by purchasing the service of fake stars

This work is supported by the National Key Research and Development Program of China No. 2018AAA0102304, and the National Natural Science Foundation of China under Grant No. 62177003. *(Corresponding author: Li Zhang.)*

DOI reference number: 10.18293/DMSVIVA22-005

[1] https://help.github.com/en/github/getting-started-with-github/saving-repositories-with-stars

[2] https://www.theregister.co.uk/2019/07/30 /would_you_star_a_github_project_for_a_free_drink/

[3] https://www.jianshu.com/p/4820cbace5c1

from an online shop, and manually select and check 240 normal users whose star repositories are owned by authoritative owners. The experimental results show that AUDetec has a high performance by achieving an accuracy of 99.86% on average.

The main contributions of this paper are as follows:

- We proposed an abnormal user detection model AUDetec, which uses the decision tree to analyze features starring_repository_number and repository_age.
- Experiment results show that AUDetec has a high performance by achieving an accuracy of 99.86% on average.

The remainder of the paper is organized as follows. Section 2 introduces the background of our study and data collection. Section 3 introduces how we extract features. Section 4 introduces the method AUDetec for detecting abnormal suers. Section 5 presents the results of the evaluation for the method AUDetec. Section 6 presents the related work. Finally, Section 7 concludes our work.

II. Background and data collection

In this section, we illustrate the background for users in GitHub to perform starring activity on repositories and the emergence of fake stars. Then, we introduce how we build the dataset.

A. Background

GitHub is a distributed version control system, providing services for individuals and teams to manage repositories via Git [5]. In GitHub, each repository has a star button for users to click. Users perform starring activity on repositories mainly to show appreciation to repositories, bookmark repositories for later retrieval and keep track of updates in news feed [1]. According to the previous work [1], three out of four developers consider the number of stars before using or contributing to repositories, and the number of repositories' stars are viewed as the most useful measure of popularity in GitHub. Furthermore, some researchers select the GitHub repositories for their study base on the number of stars of repositories [2]. Therefore, the number of stars in a repository becomes a critical metric reflecting the repository's quality and popularity.

Due to the importance of stars, and as the open-source community receives increasing recognition in companies, developers who have repositories with many stars are welcomed in the job market. Some developers use fake stars to increase the success rate of job hunting. For example, some developers obtain a large number of fake stars in a mutual star community GitStar, and successfully receive offers from big internet companies such as Alibaba. In the community GitStar, developers star repositories for other users, who then star developers' repositories in return. This community provides professional scripts for conveniently performing the starring activity, managing starring records, and reminding members to star repositories in return[4]. If a developer only receives stars from other users but does not give stars to others' repositories, this developer is forbidden by the community.

Fake stars ruin efforts that communities have made stars a valuable indicator, and bring several negative impacts in GitHub. First of all, developers often consider the number of stars before using or contributing to repositories [1]. Fake stars may mislead developers into contributing to poor-quality repositories that have many fake stars.Second, repositories with fake stars are overvalued, and developers with these repositories give interviewers false impressions, which finally affects the recognition of GitHub stars in the job market. Fake stars are also unfair for developers who work hard and well on their own repositories. Therefore, it is important to stop abusing GitHub stars and detect abnormal users who provide fake stars.

B. Data collection

The dataset is used for designing the detection method for abnormal users. In this dataset, we need to know who are actually abnormal users and normal users.

This dataset includes information of abnormal users and normal users, which is used for designing the detection method. We first need to determine a list of users who are actually abnormal and normal. and then collect their information.

Though GitHub monitors malicious activities and bans corresponding users, the list of users forbidden by GitHub is not public, and we can not directly obtain abnormal users from GitHub. Instead, we collect abnormal users by purchasing the service from a online shop which provides fake stars for repositories[5]. This shop claims that it helps users find good jobs by increasing stars of their repositories. The shop writes that accounts are manually maintained with different IP addresses, and they are used to add fake stars of repositories with the speed determined by artificial intelligence technology. In order to obtain abnormal users, we create a phishing repository named *shizheng0510/apiCrawler*. We purchased 120 fake stars for our phishing repository from the shop in April 2019. These 120 users who starred our phishing repository constitute abnormal user set.

We obtain 120 abnormal users by purchasing the service of fake stars from a online shop. In order to make comparison, we need to obtain some normal users. We find 2 repositories*git/git.github.io* and *elastic/apm-agent-nodejs*. We manually determine that these repositories do not have fake stars for following reasons. First, these repositories are owned by authoritative owners git[6] and elastic[7], who are unlikely to buy fake stars. Second, we manually check changes of their stars and find they are reasonable. The repository*git/git.github.io* was created in February 2014, and it had 132 stars. The repository *elastic/apm-agent-nodejs* was created in August 2017, and it had 287 stars. We randomly collect 120 users who star each repository, and obtain 240 users to construct the

[4] https://gitstar.com.cn/

[5] https://item.taobao.com/item.htm?spm=a1z02.1.1997525049.3.7de3782dQlNweb&id=583515543354

[6] https://git-scm.com/

[7] https://www.elastic.co/cn/

normal users set. The third author manually checks these 240 users, and make sure that they do not have abnormal activities. GitHub provides APIs for researchers to download data. We use APIs provided by GitHub to collect information of above abnormal users and normal users.

III. Feature analysis

In order to identify abnormal users who provide fake stars, we need to know differences between abnormal users and normal users. In this section, we first define features, and then perform differential analysis to compare abnormal users and normal users.

A. Features

Hudson et al. studies three dimensions of features potentially affecting the star growth of a repository, including owner dimension, repository dimension, and activity dimension [1]. According to owner dimension in the previous work [1], we consider the user dimension to describe features related to a user. In the previous work [1], the repository dimension includes features that are accessible to users on the repositories' page, and the activity dimension includes features related to recent development activity in the repository. In this paper, we combine repository dimension and activity dimension together to describe features of repositories which are starred by a user. We introduce the user dimension and repository dimension as follow.

The user dimension describes features related to a user. Features account_type, belong_organization and has_email include basic information of the user. The follower_number and the following_number features are used to reflect the user's social status. Features owning_repository_number and total_obtain_stars are considered to measure repositories owned by the user. Normal users and abnormal users may behave differently and have various user features. For example, abnormal users may be inactive and have fewer followings who follow them [6].

Besides user's features, repositories that users star may reveal evidence for detecting abnormal users. Normal users may star some good repositories which reach certain quality. However, abnormal users may star some repositories of poor quality for special purposes, such as earning money. Therefore, we consider some features of repositories which are starred by a user. All repository features are used to measure repositories starred by a user, rather than repositories owned by a user. First, we consider how users are active in starring repositories. The feature starring_time_interval means the median value of the time interval between consecutive starring activities. The feature starring_repository_number means the sum of repositories starred by the user. Second, features repository_star_number, fork_number, network_number, and subscriber_number are considered to measure the popularity of repositories which are stared by the user. fork_number only measures direct forks, while network_number measures all forks, such as forks of forks. Third, repository_age, is_fork, repository_size, has_wiki, has_page, and description_length provides basic information of repositories which are stared by the user. Finally, we measure development activities in repositories by considering their commits, contributors, git tags, releases, issues, pull requests and updates. For example, higher number of commits might indicate that the project is in constant evolution and the number of contributors, issues, and pull requests might indicate the engagement of the repository community.

In comparison with the previous work [1], we add 2 new features and delete 4 features. Previous work [1] analyzed stars from the perspective of repositories, and this paper studies stars from the perspective of abnormal users. In order to measure users' activeness in starring repositories, we add 2 features starring_time_and starring_repository_number. GitHub's API does not directly return repositories' domains, README files or the information about whether they are mirrors, and thus we do not consider these features in this paper. Some repositories are written by different programming languages [7]. GitHub only provides the main programming languages of repositories, and does not provide other programming languages which are also used in repositories. Therefore, we do not consider the feature about programming language.

B. Differential analysis

In this subsection, we compare features of normal users and abnormal users. We use the Mann-Whitney U Test to assess statistically significant difference with $\alpha = 0.05$. If the feature value shows a significant difference between normal users and abnormal users, we reserve this feature for the detection method. Otherwise, we remove this feature from the detection method of abnormal users.

We describe data collection in the subsection II-B. We compute feature values for normal users and abnormal users. and then compare their values. Table I shows average values of normal users, average values of abnormal users, and their significance difference.

As we can see from Table I, account_type and starring_time_interval appear no significant difference. All samples that we choose belong to individual types and average values of account_type for normal users and abnormal users are both 0. Therefore, account_type appears no significant difference between normal users and abnormal users, and we delete the feature account_type for the detection method. We also delete the feature starring_time_interval for the detection method, because this feature shows no significant difference between normal users and abnormal users. In Table I, other features have significant difference with $\alpha = 0.05$, and we remain them for designing the detection method of abnormal users. For example, in comparison with normal users, abnormal users star repositories which have fewer development activities, and have fewer commit_number, contributor_number, tag_number, release_number, issue_number, pull_request_number.

IV. Detection method for abnormal users

Since a user is normal or abnormal, the detection can be considered as the user classification. In this section, we intro-

TABLE I
FEATURES VALUES OF NORMAL USERS AND ABNORMAL USERS

Dimension	Feature	Average value of abnormal users	Average value of normal users	Significance difference
User	account_type	0.000	0.000	1.000
	belong_organization	0.175	0.421	<0.001
	has_email	0.000	0.492	<0.001
	owning_repository_number	6.892	126.100	<0.001
	gist_number	0.875	12.850	<0.001
	follower_number	1.417	82.567	<0.001
	following_number	2.825	344.333	<0.001
	account_age	190.833	1979.917	<0.001
	total_obtained_star	1.158	272.421	<0.001
Repository	starring_time_interval	78580.483	125588.713	0.293
	starring_repository_number	19.967	70.271	<0.001
	repository_star_number	166.625	434.988	<0.001
	fork_number	18.217	62.346	<0.001
	network_number	18.217	65.746	<0.001
	subscriber_number	1.892	26.521	<0.001
	repository_age	90.958	777.721	<0.001
	last_push_time_interval	20.092	18.513	<0.001
	is_fork	0.000	0.026	<0.001
	repository_size	1716.992	2448.175	<0.001
	has_wiki	0.787	0.770	0.017
	has_page	0.077	0.198	<0.001
	has_homepage	0.222	0.437	<0.001
	description_length	46.083	49.525	<0.001
	commit_number	25.808	184.713	<0.001
	contributor_number	1.125	8.525	<0.001
	tag_number	0.008	2.492	<0.001
	release_number	0.000	0.083	0.004
	issue_number	1.658	7.996	<0.001
	pull_request_number	0.000	0.596	<0.001

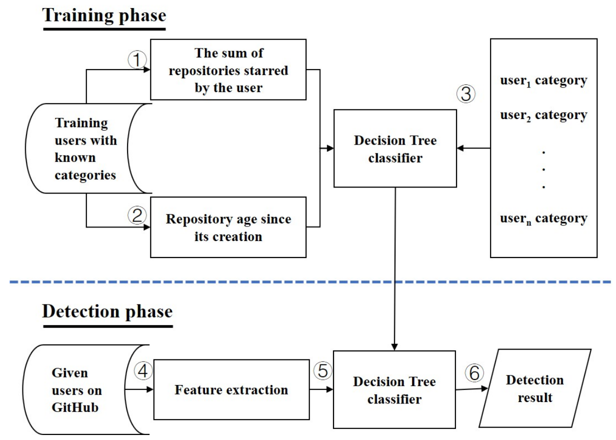


Fig. 1. Overall framework

duce our detection method AUDetec, which detects Abnormal User by the decision tree algorithm. The overall framework is presented as Figure 1. There are two phases during the whole process, namely training phase, and detection phase.

A. The sum of repositories starred by the user(Step 1)

The feature starring_repository_number is the sum of repositories starred by the user. Normal users perform starring activity on repositories that they are interested in. The number of repositories starred by the normal users results from the accumulation of their starring behavior. In contrast, abnormal users are created to accomplish their starring mission, and abnormal users do not have to star unrelated repositories. The sum of repositories starred by abnormal users may be smaller and similar because they are controlled by the same attacker. Therefore, we use the feature starring_repository_number to detect the abnormal users.

B. Repository age since its creation(Step 2)

The feature repository_age is the median value of the number of days since creation for repositories starred by the user. As shown on Table I, there is a difference in repository_age between normal users and abnormal users. The normal users tend to perform the starring activity on repositories with high-quality. These repositories generally have been created for a long time, which guarantees the numbers of contributors for maintaining the repositories. However, repositories starred by the abnormal users are generally newly created and may be urgent for a great many of stars to ”improve” the repositories' recognition. Therefore, we consider using the repository_age to detect the abnormal users.

C. decision tree classifier(Step 3)

In this subsection, we build a decision tree classifier based on the training dataset. We obtain the starring_repository_number and repository_age, and generate a two-dimensional feature vector for each user. We build a label for each vector using the value 0 or 1 to describe the user's class, in which 0 represents the normal user and 1 represents the abnormal user. The detection of abnormal users is converted to the classification of users. Then, we analyze feature vectors and build the classifier based on the decision tree. We choose the decision tree since the decision tree achieves the best performance in the detection, and it clearly provides feature weights which allows us to choose necessary features. More details are described in subsections V-C and V-D.

D. Detection phase(Step 4 to Step 6)

The detection phase is similar to the training phase. Given a user set in GitHub, we calculate values of features the starring_repository_number and repository_age (step 4). Then, we use the decision tree classifier built in the training phase to compute category probabilities (Step 5). Finally, the decision tree classifier outputs the classification result, and users in the abnormal category are identified as providing fake stars on special purpose (Step 6).

V. Experiment

In this section, we present the results of our evaluation for the proposed approach. The aim of this study is to investigate the effectiveness of AUDetec approach in detecting abnormal users. We first present the evaluation procedure and research questions. We then present our experimental results that answer these research questions.

A. Evaluation Procedure

As introduced in the subsection II-B, we collect the dataset of abnormal users and normal users. In each round, we randomly split the dataset into the training set and the test set. The training set includes 80% of users, and the test set includes other 20% of users. In order to reduce the impacts of user selection, we evaluate the performance of 10 rounds and compute average results.

A user is correctly recognized if a real normal user is identified as normal, or a real abnormal user is identified as abnormal. We use the accuracy to measure the performance in detecting abnormal users. We use N to represent the number of users in the test set, and use N_p to represent the number of users who are correctly recognized. Then the accuracy P is defined as $\frac{N_P}{N}$.

B. Research questions

We perform a study to answer the following three research questions.

RQ1: ***What is the benefit of the decision tree in detecting abnormal users?***

AUDetec uses the decision tree to detect abnormal users. We would like to investigate whether random forest achieves better performance than some other machine learning algorithms. We detect abnormal users based on SVM, Naive Bayes, and KNN, respectively. Then we compare the performance of different algorithms in detecting abnormal users.

RQ2: ***What are the weights of features in abnormal user detection?***

As shown in Table I, abnormal users are much different from normal users. We wonder whether a few features are enough to distinguish normal users and abnormal users. The decision tree provides weight for each input feature. It reflects how important a feature in the classification model. Moreover, the analysis of feature weight can help to delete redundant features. Therefore, we calculate the weights of features and describe why AUDetec only chooses features starring_repository_number and repository_age.

C. RQ1: Benefits of decision tree

TABLE II
Accuracy with different algorithms

Round	Decision tree	SVM	Naive bayes	KNN
1	98.61%	100%	98.61%	98.61%
2	100%	98.61%	98.61%	100%
3	100%	100%	98.61%	98.61%
4	100%	95.83%	98.61%	97.22%
5	100%	100%	100%	100%
6	100%	98.61%	98.61%	100%
7	100%	98.61%	98.61%	98.61%
8	100%	98.61%	98.61%	98.61%
9	100%	97.22%	98.61%	97.22%
10	100%	98.61%	97.22%	98.61%
Average	99.86%	98.61%	98.61%	98.75%

AUDetec uses the decision tree to detect abnormal users. In this subsection, we investigate the performance of different machine learning algorithms, including decision tree, SVM, Naive bayes, and KNN. The SVM classifier is a supervised classification algorithm that finds a decision surface that maximally separates the classes of interest. Naive bayes is a probabilistic machine learning algorithm based on the Bayes Theorem. KNN (K-Nearest-Neighbors) categorizes an input by using its k nearest neighbors.

Table II shows the performance of different algorithms in detecting abnormal users. Results show that all algorithms achieve high accuracy because there exists a significant difference between abnormal users and normal users. The average accuracy of detection based on the decision tree is slightly larger than the average accuracy of detection based on other machine learning algorithms. Therefore, we choose the decision tree as the classification algorithm in AUDetec.

D. RQ2: Feature weights

Table I shows that abnormal users are much different from normal users. We wonder whether a few features are enough to distinguish normal users and abnormal users. The decision tree provides weight for each input feature. It reflects how important a feature in the classification model. According to steps described in subsection IV-C, we calculate the information gain for features and then build the decision tree. For the

TABLE III
FEATURE WEIGHTS

Round	starring_repository	repository_age
1	86.06%	6.78%
2	86.66%	6.81%
3	81.20%	9.92%
4	84.06%	8.45%
5	84.06%	8.45%
6	82.05%	8.33%
7	80.16%	9.84%
8	80.94%	8.26%
9	82.49%	10.02%
10	80.86%	9.89%
Average	82.85%	8.68%

decision tree built in each round in Table II, we also compute weights of features, and describe results in Table III. Results show that the decision tree only has two features, including starring_repository_number and repository_age. Those two features are enough to detect abnormal users. Therefore, we delete redundant features and only choose these two features in the detection method AUDetec.

VI. Related work

Related work of this study could be divided into the following two parts.

A. Staring behavior

Borges et al. revealed the motivation why users perform a starring activity on a repository in GitHub and proposed a growth pattern characterization of the GitHub star based on machine learning [1]. Begel et al. showed stars are the primary consideration for whether a repository is trending or not [5].

What is more, the number of stars is a vital metric for researchers to select GitHub repositories. Ray et al. and Borges et al. used the number of stars as the metric of a project's popularity [2], [8].

Nevertheless, the above-mentioned work did not study how to detect abnormal users with fake stars in GitHub.

B. Methods to detect abnormal users

Aiming at the threats brought by abnormal users, researchers proposed various strategies to detect abnormal users with different malicious activities.

Detection based on user behave features Initial studies [3], [9] designed approaches to detect fraudulent accounts and Sybil accounts, who perpetrated scams, phishing, and malware. These algorithms depend barely on the users' behavior. Therefore, these algorithms may not work when the behavior changes due to the fraudulent accounts and Sybil accounts users alternate their actions.

Detection based on relationship graph Some studies [9], [10] detected potential Sybils accounts, which are used to introduce spam or manipulate online voting, based on the relationship graph. However, many outcomes show that detection based on a relationship graph manifests moderate accuracy, which makes it better for researchers to use this strategy as a supplementary method.

Detection based on unsupervised learning Viswanath et al. selected user behavior, and leveraged PCA to model the behavior of normal users and identify significant deviations from it as anomalous [4]. They successfully detected diverse attacker strategies which varied from fake, compromised, to colluding Facebook identities. However, this strategy consumes abundant time and memory resources since it needs vast data to train the model.

All the technics proposed above are based on other platforms such as Renren, Facebook. Our research study an approach using the machine learning method to detect abnormal users with fake stars in GitHub.

VII. Conclusion

In this paper, we propose to detect abnormal users with fake stars in GitHub. According to previous work [1], we first define features from the user dimension and repository dimension. Then we perform differential analysis and find that most of the features show a significant difference between abnormal users and normal users. Next, we propose an approach AUDetec for Abnormal User Detection, which uses the decision tree to analyze features starring_repository_number and repository_age. We evaluate the effectiveness of AUDetec based on 120 abnormal users and 240 normal users. Results show that AUDetec has a high performance by achieving an accuracy of 99.86% on average.

References

[1] H. Borges and M. T. Valente, "What's in a github star? understanding repository starring practices in a social coding platform," *Journal of Systems and Software*, vol. 146, pp. 112–129, 2018.

[2] B. Ray, D. Posnett, V. Filkov, and P. Devanbu, "A large scale study of programming languages and code quality in github," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2014, pp. 155–165.

[3] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse," in *22nd USENIX Security Symposium*, 2013, pp. 195–210.

[4] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Towards detecting anomalous user behavior in online social networks," in *23rd USENIX Security Symposium*, 2014, pp. 223–238.

[5] A. Begel, J. Bosch, and M.-A. Storey, "Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder," *IEEE Software*, vol. 30, pp. 52–66, 2013.

[6] K. Blincoe, J. Sheoran, S. Goggins, E. Petakovic, and D. Damian, "Understanding the popular users: Following, affiliation influence and leadership on github," *Information and Software Technology*, vol. 70, pp. 30–39, 2016.

[7] P. S. Kochhar, D. Wijedasa, and D. Lo, "A large scale study of multiple programming languages and code quality," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 563–573.

[8] H. Borges, A. Hora, and M. T. Valente, "Predicting the popularity of github repositories," in *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering*. Association for Computing Machinery, 2016, pp. 1–10.

[9] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, pp. 1–29, 2014.

[10] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content voting," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association, 2009, pp. 15–28.

EXplainable AI for Smart Agriculture

Andrea Cartolano[1], Alfredo Cuzzocrea[2,3]*, Giovanni Pilato[4]
[1]Department of Economical, Business and Statistical Sciences, University of Palermo, Palermo, Italy
andrea.cartolano@community.unipa.it
[2]iDEA Lab, University of Calabria, Rende, Italy
[3]LORIA, University of Lorraine, Nancy, France
alfredo.cuzzocrea@unical.it
[4]ICAR-CNR, Istituto di Calcolo e Reti ad Alte Prestazioni, Italian National Research Council, Palermo, Italy
giovanni.pilato@cnr.it

Abstract

We analyze a case study in the field of smart agriculture exploiting Explainable AI (XAI) approach. The study regards a multiclass classification problem on the Crop Recommendation dataset. The original task is the prediction of the most adequate crop according to seven features. In addition to the predictions, two of the most well-known XAI approaches have been used in order to obtain explanations and interpretations of the behaviour of the models: SHAP (Shapley Additive ExPlanations), and LIME (Local Interpretable Model-Agnostic Explanations).

Index terms— Smart Agriculture, XAI, SHAP, LIME

1 Introduction

Smart agriculture is an ideal field of application for the key concepts of industry 4.0 [32, 25, 29, 28]. Global warming and climate change have been threatening the agricultural and food production processes, impacting both contexts that are becoming more and more important to manage to avoid potentially huge losses for what concerns the cultivation of the crop. Lacks of water put the chain of food production at risk, especially in some world regions. A new approach, known as Smart Agriculture, combining traditional agriculture with artificial intelligence (AI) and autonomous systems, is then needed to tackle the challenges represented by climate change. [26]. Classifications tasks are part of the machine learning (ML) techniques, whose goal is to predict unordered discrete values. Classification can be binary, and in this case we have only two classes, or multiclass, and it is the case of our study, where we have more than two classes to predict [18]. Many classification tasks associated with smart agriculture have been proposed in the literature. However, among the different approaches, an interpretable methodology is needed since the audience is represented in the first place by farmers and agronomists. The kind of audience involved is crucial in XAI [1]: farmers and agronomists, generally speaking, are pretty skeptical in trusting AI and machine learning model predictions, especially in the presence of non-interpretable black-box models, as reported in [31]. Furthermore, the visualization of the results plays a crucial role in this field since the final users of this methodology are usually not experts in machine learning, and visual output is therefore desirable (e.g., [12, 15]). In one paper [31], a Fuzzy Rule Based-System (FRBS) has been implemented, and it is well shown that an interpretable fuzzy system can better represent the knowledge of farmers, which, in most cases, is not so precisely defined. Another approach is described in [17] where a Case-Based Reasoning (CBR) model has been developed whose aim is predicting the ideal grass growth to cultivate and making interpretable the predictions for farmers and agronomists. The problem of trust in machine learning models is critical since, without it, users will tend not to use algorithms in every field of application [27]. Because of this, interpretable approaches become essential rather than optional. Strictly related to this problem, the issue of coupling AI amd ML with *big data research challenges* (e.g., [9, 13, 8]) is really emerging at now, even in the smart agriculture context (e.g., [19, 22]). In this respect, as argued in recent studies (e.g., [14]), *performance* is a major concern to be investigated (e.g., [11, 10]).

In this paper, we tackle the problem of smart agriculture. We focus on Explainable AI (XAI) and how it allows to investigate the knowledge learned by machine learning models trained to recognize an adequate crop to cultivate. More

[*] This research has been made in the context of the Excellence Chair in Computer Engineering at LORIA, University of Lorraine, Nancy, France.

DOI reference number: 10.18293/DMSVIVA2022-008

in detail, we show that through some XAI charts, even non-machine learning experts can understand why a model predicts a particular crop for a specific observation and, overall, which are the best combinations of features that lead the models to the prediction of one class.

In particular, we show how different ML models can obtain high accuracy scores and how visualization XAI charts transform our black box models into interpretable transparent models. Shapley Additive ExPlanations (SHAP) [21] and Local Interpretable Model-Agnostic Explanations (LIME) [27] libraries have been used to get explanations and interpretations of the model. We show how both XAI packages can be advantageous in showing the behavior of models about the recommendation of the best crop to select among 22 possible classes.

Through the functions and the different kinds of plots that both libraries provide, we can collect interpretations on single predictions of the model and, particularly with SHAP's summary plot, even the patterns behind the recommendation of a single class.

2 Applying eXplainable AI to Smart Agriculture

The selection of the adequate crop to cultivate in relation to the soil characteristics and climate conditions is extremely important in smart agriculture, because it allows implementing ML models that can classify and predict which agricultural products are more likely to grow in the presence of specific input data. The problem can be seen as a classification task. There are many classification approaches in the literature. However, we need an explainable methodology for many reasons. First of all, there are some fields of application in which is extremely dangerous to be confident in predictions without an explanation, for instance in medical science where it has been proven that not interpretable models could potentially cost human lives [24]. The accuracy score is not enough to gain trust in the algorithms because a model can learn pieces of knowledge not included in the training set, and we may have data leakage [16]. Also, we may have models that, when used on real-world data, could obtain worst performances than expected, resulting in negative economic consequences. In the second place, XAI can encourage farmers and agronomists to use ML models or AI systems, allowing them to investigate the knowledge learned by the models, on the one hand, it is also possible to compare the human expertise with the ML knowledge. The issue that has been tackled is a multiclassification problem: the models have to predict an adequate crop for the field condition based on seven numeric features, respectively N, P, and K (showing the concentration values of nitrogen, phosphorus, and potassium within the fertilizer), temperature (in Celsius), rate of humidity (percentage), ph (acidity of the soil) and rainfalls. We have used the Crop Recommendation dataset [1] experimentally to investigate explanations and common patterns among different models. This dataset is made up of 2200 observations, there are no missing values, and it has been selected because of its completeness and simplicity. This last element is essential if one of the targets that it should be taken into consideration is the comparison between the knowledge learned by the algorithm and the knowledge of farmers and agronomists. For every model, the dataset has been divided in a training set (80% of observations) to train the models and a test set (20%) to obtain the predictions.

2.1 LIME

LIME [27] aims to explain the prediction function of the original complex model through a simpler linear model. An explanation is a local linear approximation of the original model. LIME is model-agnostic, which means that, regardless of the complexity of the original model, this algorithm will interpret it as a non-transparent model. The authors of LIME tried to find solutions to two common problems in machine learning: the gain of trust in the single predictions, on the one hand, and the gain of trust in the behavior of the model as a whole, on the other. Indeed, if users cannot understand why a model behaves as it does, they will tend not to use it. An explanation should also be locally faithful, which means that in the proximity area around the instance explained, the explanation model g should reply to the behavior of the original model f.

To explain a local prediction that is locally faithful the algorithm will minimize the loss function L, which include the original model f, the simpler linear model g and $\pi_x(z)$, that is the proximity measure between the instance x and the instance z, and $\Omega(g)$, that represents the complexity of the explanation model (e. g. depth of the trees in a decision tree).The simpler the model, the better for the interpretability of the explanation.

$$\boldsymbol{\xi(x) = \underset{g \in G}{\operatorname{argmin}} \quad \text{Ł}(f, g, \pi_x) + \Omega(x)} \tag{1}$$

To gain trust in the behavior of the entire model, the authors of LIME developed another algorithm, the Submodular Pick (SP). SP aims to select instances characterized by a non-redundant coverage of the area of the model, where non-redundant means that it is made up of instances with different explanations. Within a set B of instances that a human being is willing to inspect, through the SP, it is possible to obtain a $n \times d'$ explanation matrix, where n is the number of explanations selected by the SP and d' represents the interpretable features, while I is the total importance

[1] https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

of interpretable features that are contained in at least one selected instance. Non-redundant coverage is obtained by function c that for W and I computes the total importance of features contained in the set V of explanations.

$$c(V, W, I) = \sum_{j=1}^{d'} 1[\exists i \in V : W_{ij} > 0] I_j \tag{2}$$

The SP maximizes the mentioned coverage function by adding for each iteration the instances with the highest impact on coverage inside the set V.

$$Pick(W, I) = \underset{V, |V| \leq B}{\text{argmax}} \quad c(V, W, I) \tag{3}$$

2.2 SHAP

SHAP [21] is an additive feature attribution method that has its roots in Shapley values and Game Theory. Starting from the base value, the predicted value from the null model (i. e. the model without any feature), SHAP calculates the average marginal contribution of each player, a portion, or a group of features. For each observation, the sum of SHAP values of each feature is equal to the difference between the model's predicted value and the base value.

Explanation models use simplified inputs x' rather than the original ones through the following mapping function $x = h_x(x')$. The explanation function of such methods is a linear function of binary variables. SHAP calculates the contribution ϕ_i to each feature and by summing it is able to approximate the prediction function of the original model, where z' can be equal to 0 or 1 and M is the number of simplified input features.

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i + z'_i \tag{4}$$

SHAP values have three different desirable properties. Local accuracy is the first one and prescribes that the explanation model must be able to approximate the output of the original model either when $x = h_x(x')$ or $\phi_0 = f(h_x(0))$

$$f(x) = g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{5}$$

The second property, missingness, requires the missing features to have no impact on the model's output.

$$x'_i = 0 \Longrightarrow \phi_i = 0 \tag{6}$$

The third property, consistency, states that if the contribution of a simplified input, regardless of the other ones, does not also decrease, the original input should do the same.

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \tag{7}$$

for all inputs $z' \in \{0, 1\}^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$, where $f_x(z') = f(h_x(z'))$ and $z' \setminus i$ equates to set $z_i = 0$.

In order to compute Shapley values, the model has to be trained for each possible subset S of the entire set of features F. In this way, it is possible to attribute to each feature an importance value that corresponds to the contribution of each feature to the model prediction. To compute this value, a model $f_{S \cup \{i\}}$ is both trained with a particular feature and without the same one f_S. By doing so, predictions from the two different models are compared $f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)$, where x_S represents the input features in the subset S. By replying this procedure for each feature, it is possible to obtain a feature attribution ϕ_i for each observation.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \tag{8}$$

While SHAP calculates the average marginal contribution of each feature, the model-specific Tree Explainer calculates the contributions conditioned to the subset S of features [20], where S corresponds to the non-zero indexes of z' and N is the set of all input features. The expected value is $E[f(x)|x_S]$.

$$\phi_i = \sum_{S \subseteq N \setminus i} \frac{|S|!(M - S - 1)!}{|M|!} [f_x(S \cup \{i\}) - f_x(S)] \tag{9}$$

DeepExplainer, for neural networks, assumes features independence and the linearity of deep model and is based on the Deep Learning Important FeaTures (DeepLIFT) [30]. It assigns each input x_i a $C_{\Delta x_i \Delta x_j}$ value, correspondent to the effect of an input x_i set to a reference value in contrast to the original one. Through the mapping function $x = h_x(x')$, DeepLIFT converts original values into binary values, where 0 represents the input x_i taking the reference value and 1 the original value. Deep Explainer combines small components of neural network in those of the entire model by recursively passing DeepLIFT's multipliers, defined as

$$m_{\Delta x_i \Delta x_t} = \frac{C_{\Delta x \Delta t}}{\Delta t} \tag{10}$$

where Δx is the difference between the input value and the reference value, Δt describes the difference between target neuron t and reference value and $C_{\Delta x \Delta t}$ is the contribution of the two inputs.

3 Trained Models

Five different models have been implemented and three of them have been used to get explanations: an Extreme Gradient Boosting (XGB), a Multi Layer Perceptron neural network (MLP) and three different Support Vector Machines (SVM), the first one with linear kernel, the second one with a polynomial kernel and the third one with a radial basis function kernel: only the linear SVM has been selected, since it obtains the higher accuracy score compared to the other two. The reason behind the choice of these particular models was due to the different kinds of explainer that SHAP package provides: in addition to the model-agnostic KernelExplainer (used for the linear SVM), the DeepExplainer has been used for MLP and TreeExplainer specific for tree-based algorithms for XGB. As figure 2 shows each model achieves a very high accuracy score.

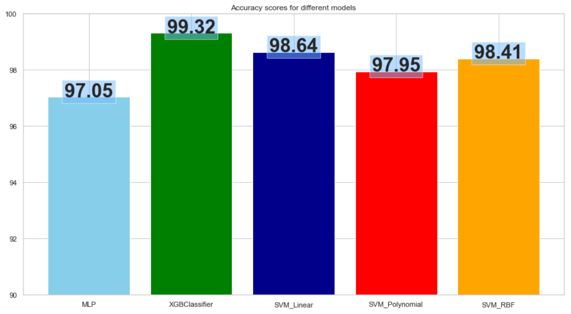


Figure 1: Accuracy scores for each model

4 XAI Charts

The functions of both SHAP and LIME packages allow to investigate on single predictions. With the method shap_values a list of 22 arrays is obtained, one for each class. In every model we have noted the tendency to misclassify the class rice in favor of class jute. The figure 3 shows the output of the XGB model for an observation in which jute has been predicted on behalf of rice.

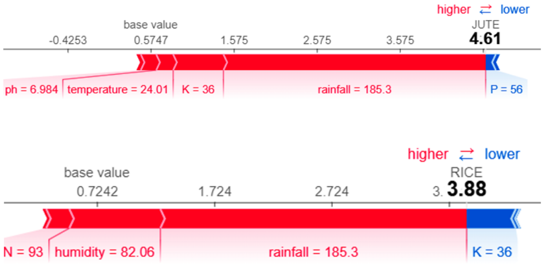


Figure 2: XGB - Force plot

With LIME's explain_instance method it is possible to obtain an easily interpretable html visualization for the same observation of figure 3, where prediction probabilities are displayed in addition to the coefficients of the features that seem to impact both positively and negatively on the prediction of one particular class.

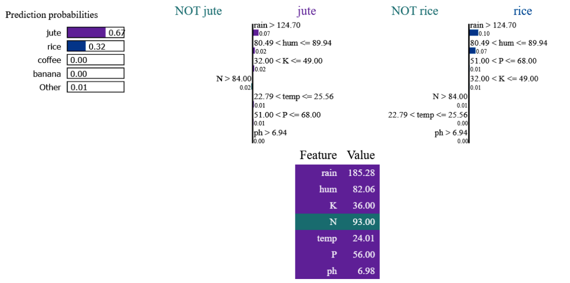


Figure 3: XGB - Explain instance

SHAP summary plot allows to display both the features with the mean biggest effect on model output and how they impact on single classes. For instance, humidity and P are the most important features for the prediction of the class apple.

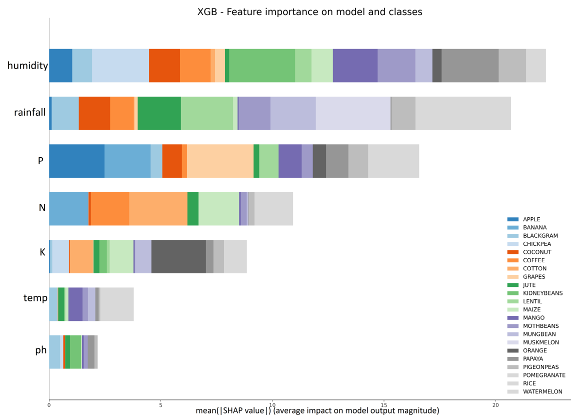


Figure 4: XGB - Summary plot

It is also possible to investigate on a single class with the same plot. It is clearly visible what has been observed in the previous plot: XGB tends to predict apple just with the contribution of P and humidity, and more precisely in presence of high values of both these features. Rainfall seems to have a slight impact whereas the other features have no impact at all for the cultivation of this crop, according to what has been learned by the XGB.

With LIME SP, as it has been mentioned in the methodologies section, it is possible to obtain a matrix W made up of n instances and d' interpretable features. In figure 5 it has been displayed the mean effect of the interpretable attributes selected by SP within a set B of 50 instances for

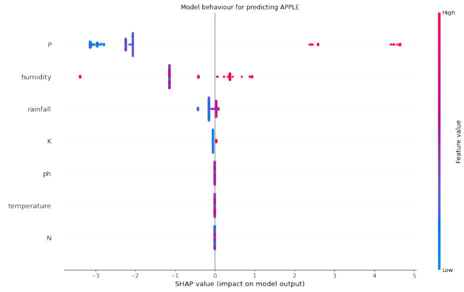


Figure 5: XGB - Summary plot for class apple

which we wanted to obtain 10 explanations. We can note that the interpretable attribute with biggest mean effect is $humidity > 89.94$ and that the attributes with poor or negative impact are related to ph and temperature, which it is consistent from what SHAP summary plot has shown.

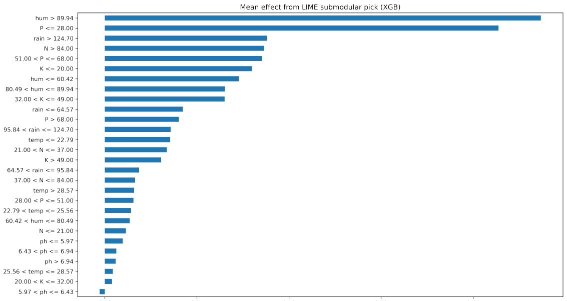


Figure 6: XGB - LIME Submodular Pick

SVM also tends to misclassify rice in favor of jute. In figure 7 the SHAP multioutput decision plot shows the model output for the 22 classes on a single observation, but just two of them are highlighted with the dashed lines. Rice is the expected class but despite it obtains positive contributions, the predicted class from the model is jute. The two classes have very similar performances until the feature humidity when a small gap starts to divide the two classes.

Another interesting visualization is given by the SHAP heatmap plot where it is visible how much each feature impacts on the instances of the test set, in addition to the global importance of each feature for the prediction of a single class. In figure 8 we reported the heatmap for class jute.

If we compare the previous heatmap with the class rice heatmap we will also have an idea of the reason why the two classes are not always correctly classified by our model. Both rice and jute are classified with the contribution of the same features: for these classes the behaviour of the model seems quite similar. However, if we look more closely we can see that rainfall attribute has a positive impact on more instances for jute rather than rice. Because of this, it is likely

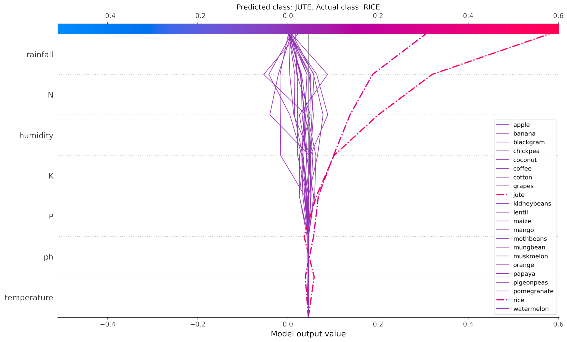


Figure 7: SVM - SHAP multioutput decision plot

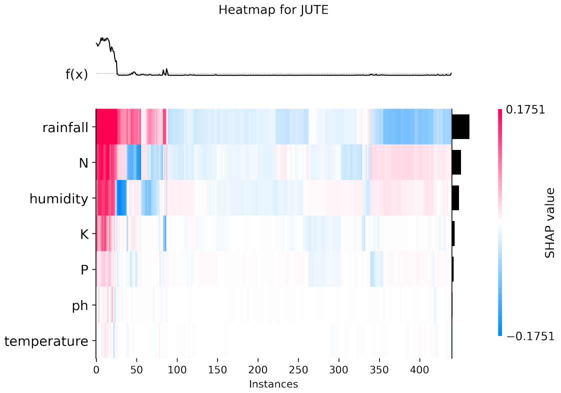


Figure 8: SVM - SHAP heatmap for jute

that jute is more easily classified than rice, in presence of similar input data.

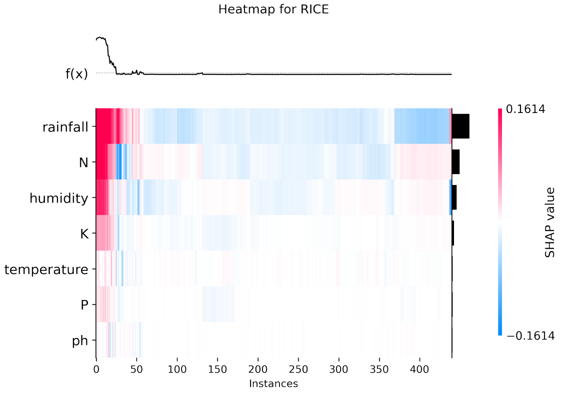


Figure 9: SVM - SHAP heatmap for rice

Unlike XGB, in SVM the feature with the biggest impact is rainfall, followed by N; temperature and ph are still the least important attributes on model output. Also, the single classes are selected differently: apple, for instance, is selected for K, rainfall, P and humidity.

The visualization of mean effect of the interpretable at-

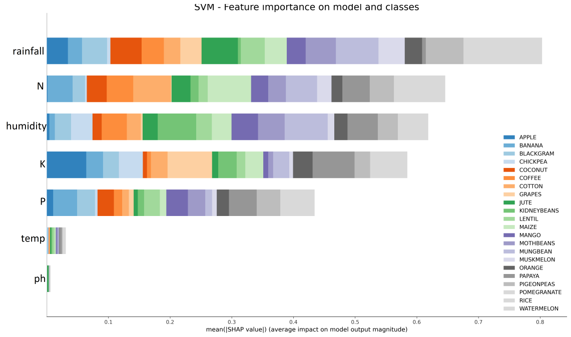


Figure 10: SVM - Summary plot

tributes selected by LIME SP is somehow similar from what summary plot shows. The interpretable attributes related to rainfall have the biggest mean effect, in particular $rainfall > 124.70$, whereas those one related to temperature and ph have poor or negative effect.

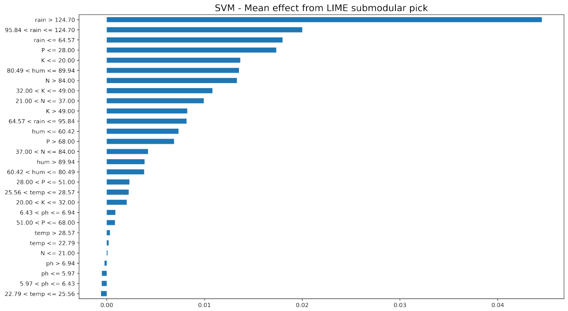


Figure 11: SVM - Submodular pick

MLP also tends to misclassify rice in favor of jute but if we look more closely at the misclassified instances we will note that with this model papaya is selected more times than jute. Indeed, papaya is a kind of crop that benefits of different values of rainfall, even lower than rice and jute, A clear representation of this pattern learnt by MLP is given by the multioutput decision plot in which for an observation that has $rainfall = 150.6$ the model output for papaya improves dramatically, unlike the expected class (jute).

By analyzing the same observation we can see how the explanation linear model approximates the behaviour of the neural network. All the interpretable attributes impact positively on the prediction of papaya but the attribute $rainfall > 124.70$ is only the third most important, whereas $32.00 < k \leq 49.00$ and $51.00 < P \leq 68.00$,

Rainfall is still the feature with the most important mean impact on model but, unlike SVM, humidity is only the 5th attribute in order of importance. The three chemical elements N, P, K are more important, with the latter that has the second biggest impact.

LIME SP seems to confirm the importance of the chemical elements: 7 out of the first 8 interpretable attributes are related to values of N, P and K. The only one that is related to a different feature is $rainfall > 124.70$, an attribute that we had already found in SVM.

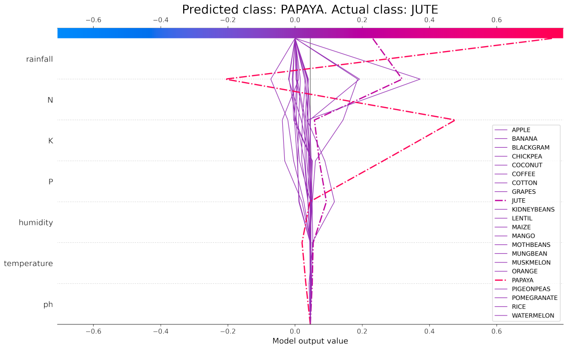
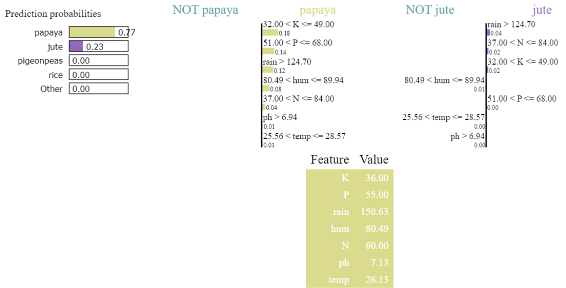


Figure 12: MLP - multioutput decision plot



Feature	Value
K	36.00
P	55.00
rain	150.63
hum	80.49
N	60.00
ph	7.13
temp	26.13

Figure 13: MLP - explain_instance

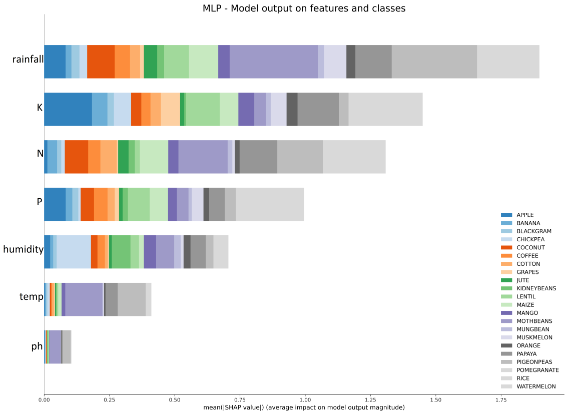


Figure 14: MLP - Summary plot

5 Conclusions and Future Work

In this paper, we have discussed how XAI can help people understand why a model selects a specific class and the logic that leads it to recommend a particular crop. The two libraries have received some criticisms, and some problems

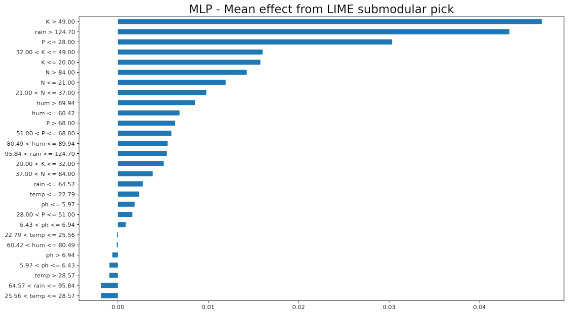


Figure 15: MLP - Submodular Pick

have to be solved to gain solidity and reliability within the scientific community, as reported in [23]. However, through the graphic options of both SHAP and LIME, we have been able to obtain explanations for single predictions and intuitive representations of how a specific model predicts a class. More in detail, we can make different observations, as it follows

Common tendencies have been found among different models: for instance, rice being misclassified in favor of jute. This is also because heavy rainfall leads to the growth of both classes. It is possible to investigate single misclassified observations. We can understand how the model behaves under the hood and why it selects a crop different from the expected one. Especially through the SHAP summary plot, we can have an intuitive idea of which class will be predicted according to the input data. This is possible even without knowing the mathematics rules included in the explanation model. We can investigate single classes and the knowledge learned by the model. Last but not least, even non-ML experts can partially understand why a model behaves. Both SHAP and LIME make the original models transparent and, regardless of their complexity, allow us to make comparisons between what a ML model learned and what farmers and agronomists know, which should always be our first concern if our target is a model that is not just accurate but also trustworthy. Future work will regard the improvement of the approach by exploiting different XAI approaches and visualization techniques, as well as using the XAI approaches in different multidisciplinary fields like computational creativity [2, 3]. Another relevant line of research consists in embedding *flexibility* in our proposed framework, for instance by adopting a semi-structured data representation format (e.g., [6, 4, 7, 5]), which may turn useful to align data with AI explanations.

Acknowledgments

This research has been partially supported by the French PIA project “Lorraine Université d’Excellence”, reference ANR-15-IDEX-04-LUE.

References

[1] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[2] A. Augello, I. Infantino, G. Pilato, R. Rizzo, and F. Vella. Introducing a creative process on a cognitive architecture. *Biologically Inspired Cognitive Architectures*, 6:131–139, 2013.

[3] A. Augello, I. Infantino, G. Pilato, R. Rizzo, and F. Vella. Creativity evaluation in a cognitive architecture. *Biologically inspired cognitive architectures*, 11:29–37, 2015.

[4] A. Bonifati and A. Cuzzocrea. Storing and retrieving xpath fragments in structured P2P networks. *Data Knowl. Eng.*, 59(2):247–269, 2006.

[5] A. Bonifati and A. Cuzzocrea. Efficient fragmentation of large XML documents. In *Database and Expert Systems Applications, 18th International Conference, DEXA 2007, Regensburg, Germany, September 3-7, 2007, Proceedings*, pages 539–550, 2007.

[6] M. Cannataro, A. Cuzzocrea, C. Mastroianni, R. Ortale, and A. Pugliese. Modeling adaptive hypermedia with an object-oriented approach and XML. In *Proceedings of the Second International Workshop on Web Dynamics, WebDyn@WWW 2002, Honululu, HW, USA, May 7, 2002*, pages 35–44, 2002.

[7] M. Cannataro, A. Cuzzocrea, and A. Pugliese. XAHM: an adaptive hypermedia model based on XML. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering, SEKE 2002, Ischia, Italy, July 15-19, 2002*, pages 627–634, 2002.

[8] A. Cuzzocrea. Overcoming limitations of approximate query answering in OLAP. In *Ninth International Database Engineering and Applications Symposium (IDEAS 2005), 25-27 July 2005, Montreal, Canada*, pages 200–209, 2005.

[9] A. Cuzzocrea. Accuracy control in compressed multidimensional data cubes for quality of answer-based OLAP tools. In *18th International Conference on Scientific and Statistical Database Management, SSDBM 2006, 3-5 July 2006, Vienna, Austria, Proceedings*, pages 301–310, 2006.

[10] A. Cuzzocrea and S. Chakravarthy. Event-based lossy compression for effective and efficient OLAP over data streams. *Data Knowl. Eng.*, 69(7):678–708, 2010.

[11] A. Cuzzocrea, F. Furfaro, and D. Saccà. Enabling OLAP in mobile environments via intelligent data cube compression techniques. *J. Intell. Inf. Syst.*, 33(2):95–143, 2009.

[12] A. Cuzzocrea, D. Saccà, and P. Serafino. A hierarchy-driven compression technique for advanced OLAP visualization of multidimensional data cubes. In *Data Warehousing and Knowledge Discovery, 8th International Conference, DaWaK 2006, Krakow, Poland, September 4-8, 2006, Proceedings*, pages 106–119, 2006.

[13] A. Cuzzocrea and W. Wang. Approximate range-sum query answering on data cubes with probabilistic guarantees. *J. Intell. Inf. Syst.*, 28(2):161–197, 2007.

[14] M. Á. Guillén, A. Llanes, B. Imbernón, R. Martínez-España, A. Bueno-Crespo, J. Cano, and J. M. Cecilia. Performance evaluation of edge-computing platforms for the prediction of low temperatures in agriculture using deep learning. *J. Supercomput.*, 77(1):818–840, 2021.

[15] B. T. Jin, F. Xu, R. T. Ng, and J. C. Hogg. Mian: interactive web-based microbiome data table visualization and machine learning platform. *Bioinform.*, 38(4):1176–1178, 2022.

[16] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

[17] E. M. Kenny, E. Ruelle, A. Geoghegan, L. Shalloo, M. O'Leary, M. O'Donovan, and M. T. Keane. Predicting grass growth for sustainable dairy farming: A cbr system using bayesian case-exclusion and post-hoc, personalized explanation-by-example (xai). In *International Conference on Case-Based Reasoning*, pages 172–187. Springer, 2019.

[18] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.

[19] P. Kumar, G. P. Gupta, and R. Tripathi. PEFL: deep privacy-encoding-based federated learning framework for smart agriculture. *IEEE Micro*, 42(1):33–40, 2022.

[20] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[21] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[22] H. Mohapatra and A. K. Rath. Ioe based framework for smart agriculture. *J. Ambient Intell. Humaniz. Comput.*, 13(1):407–424, 2022.

[23] C. Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[24] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[25] S. Qazi, B. A. Khawaja, and Q. U. Farooq. Iot-equipped and ai-enabled next generation smart agriculture: A critical review, current challenges and future trends. *IEEE Access*, 10:21219–21235, 2022.

[26] P. P. Ray. Internet of things for smart agriculture: Technologies, practices and future direction. *Journal of Ambient Intelligence and Smart Environments*, 9(4):395–420, 2017.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[28] F. K. Shaikh, M. A. Memon, N. A. Mahoto, S. Zeadally, and J. Nebhen. Artificial intelligence best practices in smart agriculture. *IEEE Micro*, 42(1):17–24, 2022.

[29] A. Sharma, M. Georgi, M. Tregubenko, A. A. Tselykh, and A. N. Tselykh. Enabling smart agriculture by implementing artificial intelligence and embedded sensing. *Comput. Ind. Eng.*, 165:107936, 2022.

[30] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[31] N. L. Tsakiridis, T. Diamantopoulos, A. L. Symeonidis, J. B. Theocharis, A. Iossifides, P. Chatzimisios, G. Pratos, and D. Kouvas. Versatile internet of things for agriculture: an explainable ai approach. In *IFIP international conference on artificial intelligence applications and innovations*, pages 180–191. Springer, 2020.

[32] G. Valecce, S. Strazzella, A. Radesca, and L. A. Grieco. Solarfertigation: Internet of things architecture for smart agriculture. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2019.

Using Visual Feedbacks in an Augmented Video-based Learning Tool

Mauro Coccoli, Ilenia Galluccio, Ilaria Torre
Department of Informatics, Bioengineering, Robotics, and Systems Science
University of Genoa, Italy
{mauro.coccoli@, ilenia.galluccio@edu., ilaria.torre@}unige.it

Abstract

In this paper, we face the problem of using video-based learning with multimedia content, which is expected to assume a prominent role in the post-pandemic world. In this respect, we have investigated the possibility of developing new services with suited visual interfaces, to further exploit its potential. Such novel services want to integrate the knowledge extracted from multimedia materials into educational applications. To do this, our approach is that of extracting the theoretical concepts included in a video lesson and describing the prerequisites relations between them, according to the knowledge base of the subject matter of the video itself. Such an addition of knowledge allows creating augmented video lessons and providing learners with new methods to browse videos and perform a non-linear navigation of their learning materials, by means of visual feedback methods. To this aim, we designed a custom web-based video player to support video-based learning, implementing these ideas. From the relevant literature we argue that this approach can be effective from an educational point of view but such effectiveness can be achieved only if the proposed video player is an easy-to-use tool, thus, we made a preliminary evaluation to assess the usability of the proposed system and results are presented.

Index terms— intelligent user interfaces, visual feedback, hypervideos, MOOCs

1. Introduction

Presently, both in the overall context of the human society, and in the specific field of education, we can observe that a "deep virtualization" process is ongoing, which is the outcome of a long evolutionary path that is leading communications and human interactions to be more and more mediated from the computer, through telematic channels and specific applications on both mobile and non-mobile *ad-hoc* devices. In particular, we want to make reference to the varied world of education and we observe that online learning has been becoming increasingly a very common practice for millions of students all over the world, which could lead to substantial and permanent modifications in current learning and teaching practices and methodologies. Indubitably, one of the main enablers of this change is the ever-increasing ease of production, recording, transmission and consumption of videos, which are more and more gaining popularity among users, at the expense of other media formats. This is true in e-learning too, where learners and teachers are following the long wave of the recent experience with Massive Open Online Courses (MOOC), which allowed finding both critical issues and advantages of using video lessons.

Now, to improve learning experiences, thus to make individuals' education and training more efficient, we are seeking novel tools that can change the way of using video lessons, hence to enhance their effectiveness. To achieve this goal, we have to face common problems that occur in the realization of learning activities based on the use of educational videos, which include, e.g.,: *(i)* their length, *(ii)* the difficulty of navigation, *(iii)* the difficulty of exploration, *(iv)* the unstructured nature of the information within, *(v)* the difficulty of highlighting the most significant parts, *(vi)* the difficulty of identifying parts purely discursive and poor in content, thus becoming useless and time-consuming. To overcome some of these limitations, the solution that we propose is based on the use of concept maps, studied before in education, which has already proved its worth and validity, demonstrating that its adoption can improve the learning experience of students, also including the ones with special needs [7, 12].

This paper presents the outcome of the activities carried on in the first stage of a larger project called *Edurell*, which aims to automatically derive concepts from within video lessons. Moreover, the extracted concepts of a specific matter are classified on the basis of their relationships in terms of prerequisites. The whole system should be considered a hyper-video service and can be managed through an *ad-hoc* visual interface.

DOI reference number: 10.18293/DMSVIVA22-004

In a broader vision, the project aims to spread the use of augmented video services within educational frameworks. In fact, despite the effectiveness of augmented video services has already been demonstrated for the students [25, 23], they are not widely adopted, due to the fact that their development is time-consuming and requires a huge manual effort of subject matter experts. More precisely, in this paper we propose the use of two types of visual feedback: (i) augmented text (from the video transcript), and (ii) a dynamic concept map (concept map flow) and we evaluate them as methods to enhance the capability of understanding videos in specific learning contexts taken in consideration as use cases.

The remainder of the paper is organized as follows: In Section 2 we present a selection of related works to support our design decisions, in Section 3 we present the proposed hypervideo model; next, Section 4 describes possible use cases and scenarios, while the results of the first evaluation through experts are outlined in Section 5. Conclusions follow in Section 6.

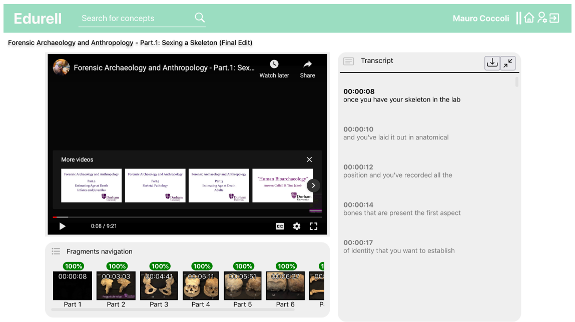
2 Related works

The use of video-based education over the last decade has been quite extensive, thanks mainly to the success of the MOOC platforms. Prior to the Covid-19 pandemic crisis, the digital transition promoted the raise of a blended learning approach [10] [14], but the pandemic situation forced institutions to move quickly to a full remote model. As a consequence, during the last two years, video production in the educational field has grown exponentially and numerous related studies have been performed to understand the point of view and the engagement of students. Recent studies show that video-based learning positively fits into the student's perception, also taking into account the variables of gender and digital inequality [24]. However, the lack of engagement is still an open issue, especially when watching a video is a passive activity and with little chance of interaction. Numerous solutions have been experimented in research to overcome this limitation, such as, e.g., the possibility of building suited environments for collaborative annotation [6], the use of self assessment quizzes to verify learning [18, 8], the adoption of interactive annotation to encourage soft skills learning [21].

The direction taken by video-based learning is that of the so-called hypervideo (HV). The definition of HV has a long history [20], as early as 2004 Zahn et. al. [33] identify the HV as a "combination of digital video and hypertext, which draws largely upon audiovisual media as central parts of their structure. They consist of interconnected video scenes containing 'dynamic' hyperlinks that are available during the course of the video scenes and that refer to further information elements (such as texts, photos and graphics)". Although the definition of HV has not yet been fully formalized, it is now common practice to refer to HV when there is some reference to interactivity such as, e.g., with control features, hyperlinks, collaboration options [5, 4, 29, 25]. Besides, the need to provide video augmentation services is closely related to another line of research that had a discrete success, owing to its positive impact on learning, that is the use of knowledge or concept maps. The research in this field has extensively tested that the application of concept maps both in different scientific domains [28, 11, 31, 9] and at different levels of education (from primary school to university) [3, 1] can have a positive impact on learners, even in the context of students with special needs and specific learning disorders [7, 12]. The way concept maps are integrated into hypervideo services is strictly correlated to the issue of information visualization and to the importance of the content presentation, in order to have a certain effectiveness, in the consolidated perspective that even in the educational field the learner can be considered as a prosumer [16]. Many projects based on a data driven approach have explored the different possibilities for improving the navigation experience, such as data-enhanced transcript search and keyword summary, automatic display of relevant frames, a visual summary representing points with high learner activity [17], non linear consumption of videos using personalized fragment navigation [32], exploration of e-learning contents via small screens [27]. The idea of using concept maps to support video navigation is already present in works at the beginning of the new millennium [13] but, at the best of the authors' knowledge, the novelty of this contribution can be identified in the possibility of automating the creation of concept maps, hopefully going in the direction of creating an on-the-fly service. In our project, the concept map supporting the video must be regarded as "interactive" and not static, as it is strictly related to the contents presented within the video, which are automatically highlighted within a relevant graph of knowledge, underlying the system and describing the contents within the video lesson.

3 Description of the system

The proposed application is an enhanced viewer for students watching video lessons, which offers additional functionalities in side panels, to enrich the learners' experience. In fact, aside the main video player, one can find (i) the transcript of the speech with important concepts highlighted (see Fig. 1), and (ii) a knowledge graph representing the prerequisites relations between such concepts (see Fig. 2). Put brief, this acts as a contextual help for the concepts explained by the lecturer who recorded the video, allowing students to navigate in the subject on the basis of their individual level of knowledge. To this aim, the knowledge



00:00:08
once you have your skeleton in the lab

00:00:10
and you've laid it out in anatomical

00:00:12
position and you've recorded all the

00:00:14
bones that are present the first aspect

00:00:17
of identity that you want to establish

Figure 1: The video player and the transcript panel

graph panel shows the prerequisites that the student has to know in order to get the current concept. The same panel also provides anchor points to that concept in the video. The system is made available online on a local development server and anyone can access upon free registration to test its functionalities. Guest users who register can enhance the current library of video lessons by providing YouTube links to their lessons, and exploit the functionalities of the system creating their own graphs and maps, making the relevant annotations.

The major contribution with respect to the literature is that, since concepts evolve as the video flows (as their explanation goes deeper), they are initially presented with lower complexity, resulting in a contextual help, which shows a simple knowledge graph. Later on in the video, the same concepts may be deepened with additional notions (concepts) that will have been expressed in the meantime. This is reflected in the dynamic contextual help, where the graph knowledge is progressively enriched and becomes more complex (see Fig. 3). Thus, concepts do not have a static set of prerequisites during the whole video and, consequently, their contextual help evolves dynamically, in accordance to the video flow. This has relevant potential applications towards personalized contextual help, when the domain knowledge graph is matched against the knowledge graph of the learner.

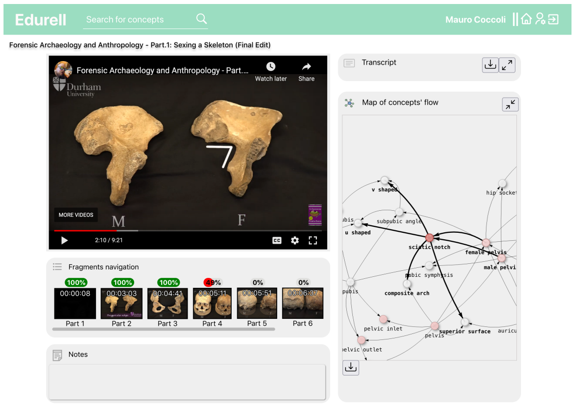


Figure 2: The video player and the knowledge graph panel

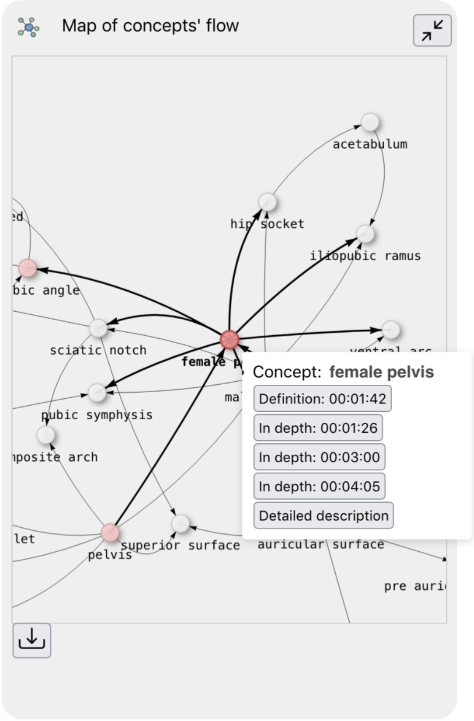


Figure 3: The dynamic contextual help mechanism, over the knowledge graph with prerequisites

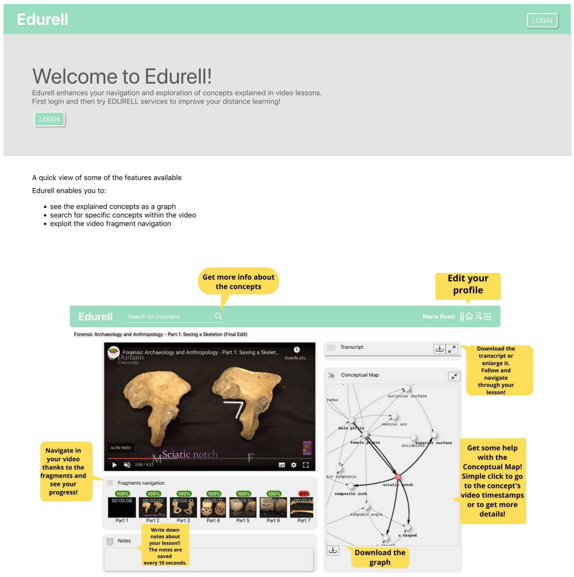
4 Use cases and scenarios

To assess the validity of the proposed solution, we identified different scenarios, outlined in the description of possible user-stories. Specifically, we considered the cases depicted in the following situations.

4.1 First time viewing

Imagine you are a bachelor's degree student in Archaeology. Within your course of study, face-to-face courses, online lessons and courses on MOOC platforms (as additional activities to get credits) are provided. Then you decide to attend the "Forensic Archaeology and Anthropology Course" in autonomous mode. The Edurell platform, with its hypervideo functionalities, will provide support to follow the course on your own. In this case, specific features of the application (e.g., browsing with fragment navigation, see below) can improve your browsing experience of a video

http://130.251.47.105:5000



Welcome to Edurell!

Edurell enhances your navigation and exploration of concepts explained in video lessons.
First login and then try EDURELL services to improve your distance learning!

A quick view of some of the features available

Edurell enables you to:

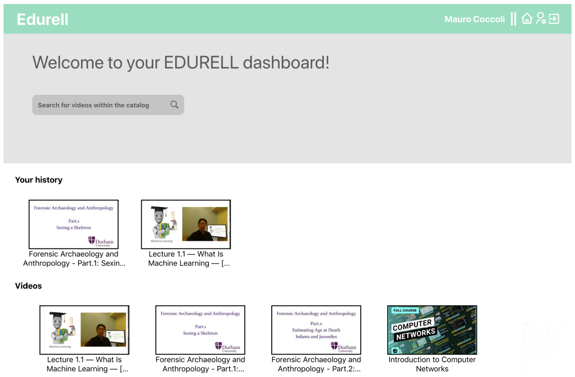
- see the explained concepts as a graph
- search for specific concepts within the video
- exploit the video fragment navigation

Get more info about the concepts
Edit your profile
Edurell
Search for concepts
Maria Rossi
Forensic Archaeology and Anthropology - Part.1: Sexing a Skeleton (Final Edit)
Forensic Archaeology and Anthropology - Part.1: Sexing a Skelet...
Durham University
Guarda più...
ALTRI VIDEO
Sciatic notch
M
F
2:09 / 9:21
Transcript
Download the transcript or enlarge it. Follow and navigate through your lesson!
Conceptual Map
ramus
auricular surface
ventral arc
male pelvis
female pelvis
childbirth
superior surface
bic symphysis
sciatic notch
composite arch
subpubic angle
u shaped
Get some help with the Conceptual Map! Simple click to go to the concept's video timestamps or to get more details!
Navigate in your video thanks to the fragments and see your progress!
Fragments navigation
100%
00:00:08
Part 1
100%
00:03:03
Part 2
100%
00:04:41
Part 3
100%
00:05:11
Part 4
100%
00:05:51
Part 5
100%
00:06:37
Part 6
85%
00:07:44
Part 7
Notes
Write down notes about your lesson!! The notes are saved every 10 seconds.
Download the graph

Figure 4: Screenshot of the Edurell login screen

with educational content (intra-video issues). Specifically, the tasks to be performed are:

- Login
  1. sign-into the Edurell platform or sign-up creating a new account. The login screen is shown in Figure 4 and presents a preview of the internal system with a relevant description in the text balloons;
  2. Look for the video you want with the Search feature or find the video in your dashboard; Figure 5 shows the welcome page with the browsing history of the user and a list of other videos available within the system. Access the first lesson of the "Forensic archaeology and anthropology" course;
  3. After creating your profile and logging in, select the video in question.



Welcome to your EDURELL dashboard!

Your history

Forensic Archaeology and Anthropology - Part.1: Sexin...

Lecture 1.1 — What Is Machine Learning — [...

Videos

Lecture 1.1 — What Is Machine Learning — [...

Forensic Archaeology and Anthropology - Part.1:...

Forensic Archaeology and Anthropology - Part.2:...

Introduction to Computer Networks

Figure 5: Screenshot of the users' home screen of the Edurell application

- Browse the video by using the Fragment Navigation feature. Actions required are:
  1. Follow the lesson of the video-course. At the first viewing of the video lesson, you may need some features to be able to review unclear parts of the video or deepen a specific concept. Instead of random scrolling the video, jumping from one point to another without a criterion, with fragment navigation you can reach effortlessly the more useful portions of the video;
  2. Start the video and reach minute 5:30;
  3. Browse the video and reach the fragment of the video that explains the "Preauricular sulcus";
  4. Now, return to the exact moment you left the video during linear viewing.

- Browse the video by using transcript: continue watching the video, but you will be able to use the transcript to carry out your research on the concepts. Actions required are:
  1. Expand the transcript panel;
  2. Reach minute 4:35 of the video through the transcription panel;
  3. Look inside the transcription for the concept "Femoral";
  4. Reduce the size of the Transcript panel.

- Browse the video with the graph view: you are following the course and now you want to exploit the knowledge graph. The panel on the right of the screen will allow you to take advantage of numerous features to browse the video and deepen some concepts. Actions required are:
  1. Expand the Interactive Knowledge Graph panel;
  2. Follow the video and write in the notes panel what are the prerequisites of the "Sciatic notch" concept (appearing at minute 2:30);
  3. Guess what is the meaning of the edges in the Knowledge Graph?
  4. Go back to minute 1:08 (in the way you prefer, i.e., with the transcript or with the navigation bar), which concepts are red coloured in the graph? Write them in the notes panel. Why are they colored?

5. Click on the "pelvis" node to see what happens on the navigation bar;
6. Click on the red dot. Guess what's the meaning of the red hotspot?
7. Now, click on the yellow dot. Guess what's the meaning of the yellow hotspot?
8. Reduce the size of the Graph panel.

- Concept search: After having watched the complete video you may want to deepen some concepts. Now you can take advantage of the Concept search functionality. Actions required are:
  1. Use the bar for the concept search, type the name of a concept "sciatic notch";
  2. Once the concept is found and the graph displayed, use the filters to obscure the parts of the graph you are not interested in, in order to show on the screen only the concept you are looking for and its prerequisites;
  3. Now, use filters to show only the concept you are looking for and all the nodes that depend on it;
  4. Within the Concept information Panel find the information relating to the minute in which the concept is explained;
  5. Click on this link;
  6. Return to the video lesson.

4.2 Second time viewing

In the same scenario as before, you have already followed all the videos of the course and now you want to browse again the contents of video number 1 because you need to review some concepts. The tasks to be performed are:

- Login. Actions required are:
  1. Log-into the Edurell platform with your registered account;
  2. From within your dashboard access to the "Forensic archaeology and anthropology - Part 1. Sexing a skeleton" video.
- Browse the video: Start following the video lesson. Since you have already seen the video, now you want to deepen some more difficult concepts. Actions required are:
  1. Browse the video exploiting the Fragment Navigation and reach the "Preauricular Sulcus" fragment;
  2. Expand the Interactive Knowledge Graph panel;
  3. Follow the video and write in the notes panel what are the prerequisites of the "Acetabulum" concept;
  4. Click on the "Acetabulum" node and take advantage of the functionality to click on the occurrence of "Acetabulum".
- Concept search. Actions required are:
  1. Use the bar for the concept search, start by typing a concept you want to learn more;
  2. Use the filters to obscure the parts of the graph that do not interest us, in order to show on the screen only the concept you are looking for and its prerequisites;
  3. Use filters to show only the concept you are looking for and all the nodes that depend on it;
  4. Exploiting the Concept Information Panel click on the link of another video lesson of the same course, within which the concept you are looking for is explained;
  5. Take advantage of all the intra-video features to deepen the concept you wanted to understand better;
  6. Go back to the lesson N.1.

4.3 Third scenario

You follow the "Forensic archaeology and anthropology - Part 1. Sexing a skeleton" video on Youtube, without any hypervideo support. Afterwards, you will follow the "Forensic Archaeology and Anthropology - Part.2: Estimating Age at Death - Infants and Juveniles" video, within the Edurell platform, taking advantage of hypervideo functionalities. We are wondering which features allow the student to better understand the concepts and the relationships between them in an educational video. The tasks to be performed are:

- Follow the video course without hypervideo functionalities. Actions required are:
  1. Start following the "Forensic archaeology and anthropology - Part 1. Sexing a skeleton" video;
  2. Follow the video till the end, without interruption;
  3. After watching the entire video, try to provide a description of "Preauricular sulcus" concept, writing it within the notes panel;
  4. Try to write in the notes panel what concepts you need to know to understand the "Preauricular sulcus" concept;

5. What are the more advanced concepts of which Preauricular Sulcus can be considered a propaedeutic concept? Write them in the notes panel.

- Follow the video course with knowledge graph visualization. Actions required are:

  1. Login into Edurell platform;
  2. From within your dashboard access to "Forensic Archaeology and Anthropology - Part.2: Estimating Age at Death - Infants and Juveniles" video;
  3. Browse the video and reach the fragment **x**;
  4. Take advantage of Concept search to deepen your knowledge about specific concepts;
  5. After watching the entire video, try to provide a description of "Preauricular sulcus" concept, writing it within the notes panel;
  6. Try to write in the notes panel what concepts you need to know to understand "Preauricular sulcus" concept;
  7. What are the concepts of which Preauricular Sulcus can be considered a propaedeutic concept? Write them in the notes panel.

5 Evaluation through experts

The novel concept of the interactive knowledge graph representation that we introduced, required the realization of suited functionalities induced by such a new hypervideo interpretation. To enable users navigating videos according to a variety of criteria, we designed a suited user interface (UI) to fully exploit such new capabilities.

Specifically, the proposed UI merges different canvases (see Fig. 1 and Fig. 2), i.e.,

- the player for the main video;
- a frame for the video transcripts;
- an area to access the interactive knowledge graph;
- a bar for navigating the video through indexed fragments;
- a suited space for students' to take notes.

Moreover, the progress bar of the video player was enhanced with some markers in correspondence of video highlights and links to the concepts represented in the knowledge graph.

To assess the validity of such a solution, we must evaluate both the effectiveness of the proposed process for knowledge management and discovery, and the usability of the UI that we designed to this aim. Hence, for the latter, we decided to perform some usability tests based on experts' reviews. In this context, we observe that the definition of a specific set of usability heuristics would be required, tailored to the specific domain. This is due to the fact that, by their nature, traditional heuristics will not be able to evaluate the specific characteristics of our particular applications with its peculiarities related to the specific domain of education. For this reason, a new set of heuristics specifically thought for the application domain of e-learning with hypervideo will be designed, after a first phase of experimentation, devoted to identify possible causes of general usability problems.

For now, we will rely on the standard set of usability heuristics, based on the renown Nielsen heuristics model [22].

Recalling that usability is typically defined as the "ability to be used" [2] and, therefore, there can be no mathematical methods to make rigorous and accurate measurements, let us consider the case in which it is assessed through a series of usability inspections or usability tests. In this respect, one of the most widely used techniques is to carry out a heuristic evaluation to find any usability problems. This method is based on the so-called "heuristic principles" or "usability heuristics" to evaluate usability. As mentioned, each specific domain should have an adequate set of usability heuristics since the more generic or traditional ones will not be able to correctly evaluate the specific characteristics of the different types of software and applications.

Making reference to the ISO 9241-11 standard, we can give an accepted definition at international level on what is usability and its application in different fields of application [2]. In practice, this standard defines usability such as "the extent to which a product can be used by specified users to achieve specified objectives with effectiveness, efficiency and satisfaction in a specific context of use" [15]. Furthermore, we point out that there are several potentially ambiguous terms related to usability, such as, e.g., effectiveness, efficiency and completeness, and for all of these, we will use the definitions given in the ISO standard, which are as follows [15]:

- User: person who interacts with the product;
- Objective: expected result;
- Effectiveness: accuracy and completeness with which users obtain specified goals;
- Efficiency: resources spent in relation to accuracy;
- Completeness: with which users reach the objectives;
- Satisfaction: freedom from discomfort and positive attitudes towards the use of the product;

- Context of use: users, activities, equipment (hardware, software and materials), and the physical and social environments in which a product is used.

However, there is still no generally accepted definition of usability, since its complex nature is difficult to describe in a definition [19] and also the mentioned standard is still under review, in order to include new lessons learned on usability since 1998 and new elements that have emerged in relation to the very concept of usability [26]. Despite that, there is a general agreement about the usefulness of adopting the heuristic evaluation method to identify *a priori* any usability problems before performing extensive usability tests with final users [30]. We followed this approach, according to the model proposed by Nielsen and Molich [22] involving usability experts who inspect the product interface for possible usability issues. The authors conducted a heuristic study based on the evaluation of 5 experts [30]. The experts were provided with the description of the scenarios and left free to use the web application as they preferred. Then they were asked to follow Nielsen heuristics, that we are going to recall in the following, for the sake of the readers' comfort. After collecting the reviewers' remarks and concerns with the current release of the system, we analyzed them carefully. Below we summarize their impressions as follows, where we present the emerged issues and possible actions to counteract the highlighted problems.

5.1 Results from the experts' evaluation

1. Visibility of the system status. – *The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.*

   Issues: At the first attempt, the platform does not provide information nor a catalog neither a preview of the available videos. Users must search the videos within the system through a search bar and then, when typing, automatic suggestions appear. Only when you are registered and have already watched some videos, recent videos are shown in "your history". A short description is missing in general, and for watched videos, some more information should be shown such as, e.g., watched, watching, or even progress within the timeline.

   Suggestions: Such an issue can be fixed by adding a visual catalog on the start screen, presenting videos as cards with short synopsis and making search available as a second choice. Furthermore, users should have personalised home pages withe their favourites and most popular videos (possibly per category), as well as the recently seen ones. This will also resolve the problem of having an empty starting page.

2. Match between system and the real world – *The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system oriented terms. Follow real world conventions, making information appear in a natural and logical order.*

   No issues were reported by none of the reviewers, comments agreed that the used language is not too technical, nor ambiguous, neither chaotic.

3. User control and freedom – *Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.*

   Issues: escape routes are missing after search results are presented. Suggestions: an emergency exit should be provided by adding a "This is not what you were looking for?" on the results page. Also, a "Back to previous page" command should be helpful.

4. Consistency and standards – *Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.*

   Issues: The navbar includes a "Home" button but the logo is not a link to the home page.

   Suggestions: While the "Home" button is still useful for a certain class of users, the link to the logo should be added.

5. Error prevention – *Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error prone conditions or check for them and present users with a confirmation option before they commit to the action.* Issues: buttons in the navbar are too much near to each other

   Suggestion: redesign the navbar and adding a hover effect on suggested videos, to clarify the current position.

6. Recognition rather than recall – *Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.*

   No issues were reported by none of the reviewers

7. Flexibility and efficiency of use – *Accelerators unseen by the novice user may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.*

Issues: given the small number of functionalities, such feature is completely missing, apart for the chronology in home page.

8. Aesthetic and minimalist design – *Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.*

   No issues were reported by none of the reviewers

9. Help users recognize, diagnose, and recover from errors – *Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.* Issues: such feature is completely missing. Suggestion: it is good practice to design tools keeping in mind that errors can occur at every time, e.g., when selecting a video, it should be made possible to make changes or deleting from own history.

10. Help and documentation – *Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.* Issues: such feature is completely missing

Besides: exporting data from the platform is possible in the `json` format. Depending on the browser/OS combination, it may happen that the file is not downloaded, yet visualized on the screen, which results in a blank page. Moreover, in the current release, not all of the functionality are fully available, hence, this assessment is only partial.

6 Conclusions

In conclusion, we can observe that the visual feedback methods has been designed with the goal of improving video-based learning by providing a structure to video content. By increasing the immediate understanding, we could expect an improvement in the efficiency of the learning process. However, further research is needed to make a more precise assessment on the real usability of this tool. Results collected from this first round of evaluation have given us good advises to improve the overall functionalities of the system.

References

[1] A. Bes-Piá, B. T. Encarna, and M. J. Muñoz-Portero. Different applications of concept maps in higher education. *Journal of Industrial Engineering and Management (JIEM)*, 4(1):81–102, 2011.

[2] N. Bevan, J. Carter, and S. Harker. Iso 9241-11 revised: What have we learnt about usability since 1998? In M. Kurosu, editor, *Human-Computer Interaction: Design and Evaluation*, pages 143–151, Cham, 2015. Springer International Publishing.

[3] M. Birbili. Mapping knowledge: Concept maps in early childhood education. *Early Childhood Research & Practice*, 8(2):n2, 2006.

[4] A. A. Cattaneo, H. van der Meij, C. Aprea, F. Sauli, and C. Zahn. A model for designing hypervideo-based instructional scenarios. *Interactive learning environments*, 27(4):508–529, 2019.

[5] A. A. Cattaneo, H. van der Meij, and F. Sauli. An empirical test of three instructional scenarios for hypervideo use in a vocational education lesson. *Computers in the Schools*, 35(4):249–267, 2018.

[6] M. A. Chatti, M. Marinov, O. Sabov, R. Laksono, Z. Sofyan, A. M. F. Yousef, and U. Schroeder. Video annotation and analytics in coursemapper. *Smart Learning Environments*, 3(1):1–21, 2016.

[7] S. Ciullo, T. S. Falcomata, K. Pfannenstiel, and G. Billingsley. Improving learning with science and social studies text using computer-based concept maps for students with disabilities. *Behavior Modification*, 39(1):117–135, 2015.

[8] S. Cummins, A. R. Beresford, and A. Rice. Investigating engagement with in-video quiz questions in a programming course. *IEEE Transactions on Learning Technologies*, 9(1):57–66, 2015.

[9] B. J. Daley and D. M. Torre. Concept maps in medical education: an analytical literature review. *Medical education*, 44(5):440–448, 2010.

[10] C. Dziuban, C. R. Graham, P. D. Moskal, A. Norberg, and N. Sicilia. Blended learning: the new normal and emerging technologies. *International journal of educational technology in Higher education*, 15(1):1–16, 2018.

[11] G. W. Ellis, A. Rudnitsky, and B. Silverstein. Using concept maps to enhance understanding in engineering education. *International Journal of Engineering Education*, 20(6):1012–1021, 2004.

[12] M. Fesmire, M. C. Lisner, P. R. Forrest, and W. H. Evans. Concept maps: A practical solution for completing functional behavior assessments. *Education and Treatment of Children*, pages 89–103, 2003.

[13] N. Guimarães, T. Chambel, J. Bidarra, et al. From cognitive maps to hypervideo: Supporting flexible and rich learner-centred environments. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 2(2):1–7, 2000.

[14] S. Hrastinski. What do we mean by blended learning? *TechTrends*, 63(5):564–569, 2019.

[15] I. Iso. 9241-11: 1998 ergonomic requirements for office work with visual display terminals (vdts)–part 11: Guidance on usability. *Geneve, CH: ISO*, 1998.

[16] N. Izotova, M. Klimenko, and E. Nikolaenko. Information visualization in context of modern education megatrends. In *E3S Web of Conferences*, volume 284, page 09011. EDP Sciences, 2021.

[17] J. Kim, P. J. Guo, C. J. Cai, S.-W. Li, K. Z. Gajos, and R. C. Miller. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings*

of the 27th annual ACM symposium on User interface software and technology, pages 563–572, 2014.

[18] G. Kovacs. Effects of in-video quizzes on mooc lecture viewing. In *Proceedings of the third (2016) ACM conference on Learning@ Scale*, pages 31–40, 2016.

[19] J. R. Lewis. Usability: Lessons learned … and yet to be learned. *International Journal of Human–Computer Interaction*, 30(9):663–684, 2014.

[20] C. Locatis, J. Charuhas, and R. Banvard. Hypervideo. *Educational Technology Research and Development*, 38(2):41–49, 1990.

[21] A. Mitrovic, V. Dimitrova, L. Lau, A. Weerasinghe, and M. Mathews. Supporting constructive video-based learning: requirements elicitation from exploratory studies. In *International Conference on Artificial Intelligence in Education*, pages 224–237. Springer, 2017.

[22] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, page 249–256, New York, NY, USA, 1990. Association for Computing Machinery.

[23] X. Niu, J. Zhang, K. M. Xu, and X. Wang. The impact of productive failure on learning performance and cognitive load: Using hypervideo to facilitate online interactions. In *2021 International Conference on Advanced Learning Technologies (ICALT)*, pages 30–32. IEEE, 2021.

[24] D. Pal and S. Patra. University students' perception of video-based learning in times of covid-19: A tam/ttf perspective. *International Journal of Human–Computer Interaction*, 37(10):903–921, 2021.

[25] M. Perini, A. A. Cattaneo, and G. Tacconi. Using hypervideo to support undergraduate students' reflection on work practices: a qualitative study. *International Journal of Educational Technology in Higher Education*, 16(1):1–16, 2019.

[26] D. Quiñones and C. Rusu. How to develop usability heuristics: A systematic literature review. *Computer Standards & Interfaces*, 53:89–122, 2017.

[27] S. Rabiger, T. Dalkılıç, A. Doğan, B. Karakaş, B. Türetken, and Y. Saygın. Exploration of video e-learning content with smartphones. 2020.

[28] A. Regis, P. G. Albertazzi, and E. Roletto. Concept maps in chemistry education. *Journal of Chemical Education*, 73(11):1084, 1996.

[29] F. Sauli, A. Cattaneo, and H. van der Meij. Hypervideo for educational purposes: a literature review on a multifaceted technological tool. *Technology, pedagogy and education*, 27(1):115–134, 2018.

[30] W.-s. Tan, D. Liu, and R. Bishu. Web evaluation: Heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics*, 39(4):621–627, 2009.

[31] E. Van Zele, J. Lenaerts, and W. Wieme. Improving the usefulness of concept maps as a research tool for science education. *International Journal of Science Education*, 26(9):1043–1064, 2004.

[32] G. Verma, T. Nalamada, K. Harpavat, P. Goel, A. Mishra, and B. V. Srinivasan. Non-linear consumption of videos using a sequence of personalized multimodal fragments. In *26th International Conference on Intelligent User Interfaces*, pages 249–259, 2021.

[33] C. Zahn, B. Barquero, and S. Schwan. Learning with hyperlinked videos—design criteria and efficient strategies for using audiovisual hypermedia. *Learning and Instruction*, 14(3):275–291, 2004.

Attention-based mechanism for text detection in indoor scenes

Keman Zhang, Yang Zou[*], Xiaoqin Zeng, Xiangchen Wu, Lixin Han
Institute of Intelligence Science and Technology, School of Computer and Information,
Hohai University, Nanjing, China
{zkm, yzou, xzeng, wxc}@hhu.edu.cn, lixinhan2002@hotmail.com

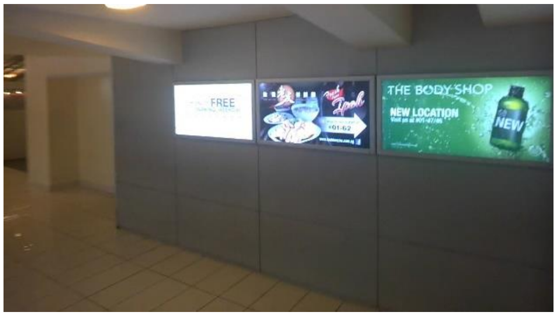
***Abstract*—Indoor scenes include the interior scenes of shopping malls, supermarkets, subways, and other buildings. Indoor scene text detection is characterized by complex background, small text target, and low image clarity, which is a difficult problem in the field of scene text detection. This paper proposes a text detection model incorporating a pixel-level attention mechanism. The model exploits Resnet18 as the backbone network for feature extraction, combines the attention mechanism with the pyramid structure for feature fusion, and introduces differentiable binarization for the prediction of text boxes. The lightweight backbone network of the model reduces computational complexity and detail loss, the pyramidal attention mechanism effectively promotes multi-scale feature fusion and enhances the ability of the model to obtain accurate location information of small targets, and the differentiable binarization facilitates distinguishing target texts from complex backgrounds, thus improving the performance of text detection in indoor scenes. Experimental results on the publicly available dataset ICDAR-2015 show that the proposed model improves the accuracy, recall, and average precision by 0.4%, 2.1%, and 1.3%, respectively, compared with DBnet, a representative model in this regard.**

Keywords-text detection; indoor scenes; attention mechanism; differentiable binarization

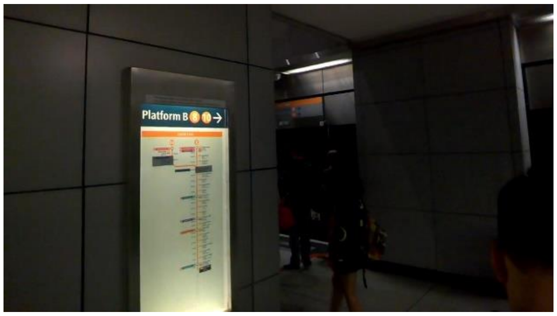
I. INTRODUCTION

Indoor scenes include interior scenes of buildings such as shopping malls, supermarkets, and subway stations. Nowadays, people are often working, studying, shopping and doing other activities indoors. The text in indoor scenes contains a lot of information, and usually plays an important role in determining locations and guiding directions. Therefore, it is of great significance to detect indoor text accurately and efficiently.

(a)



(b)



(c)

Figure 1. Text in indoor scenes.

Compared with the scene text with large and clear text targets in a simple background such as commodity signs, road signs, etc., indoor scene text has the characteristics of complex and messy background, small text targets, blurred images, and frequent lighting changes, as shown in Fig. 1. From these characteristics, it can be seen that there are two main reasons affecting the accuracy of indoor scene text detection: (1) The target texts usually occur in a complex background, which is a large obstacle to the correct segmentation when segmenting objects and backgrounds. (2) Since the sizes of target texts are usually small, their available features appear in a shallow feature map, and more and more detail information will be lost as the neural network is deepened [1].

In recent years, various types of text detection methods have emerged. Literatures [2], [3], [4], [5], [6], [7], [8], etc. exploit region proposal-based methods for text detection, whereas literatures [9], [10], [11], [12], [13], [14], etc. utilize image segmentation-based methods for text detection. Although these

[*] Corresponding author: yzou@hhu.edu.cn (Y. Zou)

DOI reference number: 10.18293/DMSVIVA22-011

text detection methods have achieved good detection results on various text datasets, they have not been adjusted or optimized for the characteristics of indoor scene text. All in all, both region proposal-based and image segmentation-based methods have drawbacks in handling text images in indoor scenes, and there is room for further improvement.

Aiming at the characteristics of indoor scene text, this paper proposes a text detection framework incorporating pixel-level attention mechanisms. The framework utilizes Resnet18 as the backbone network for feature extraction, combines the Feature Pyramidal Attention module and Global Attention Upsample module for feature fusion, employs differentiable binarization to binarize the prediction results of feature maps and output text frames. The lightweight feature extraction network improves the detection speed while reducing the loss of representational information. Feature Pyramid Attention helps recover the target location details of high-level features from the pixel level, and Global Attention Upsample acquires global context information of high-level features and low-level features for fusion. The proposed model can reduce the interference of complex background on text, enhance the target location information, and have better capability of detection for indoor scene text compared with some popular models in this regard, as indicated by the experimental results on the publicly available dataset ICDAR-2015.

II. Related Work

In recent years, a variety of techniques for text detection have been developed. Traditional methods for text detection are generally based on sliding windows and strongly connected branches, which require the manual design of features that is tedious and inefficient, and cannot meet the needs of text detection in complex scenes. With the emergence and development of deep learning, deep neural networks are widely used in text detection because of their strong capability of nonlinear fitting. At present, text detection technology based on deep learning has become a hot research direction with the advantages of accuracy and efficiency, and the mainstream methods for text detection can be divided into two categories, which are based on region suggestion and image segmentation, respectively.

Region proposal-based text detection generates candidate textboxes or candidate connected component boxes, and utilizes bounding box regression to correct the originally suggested regions so as to make the text box coordinates more accurate. FCRN [2], TextBoxes [3], SLPR [4], DMPNet [5] adopt convolutional neural network to generate text candidate boxes, and improve the design and generation of candidate boxes to enhance the ability of text detection in various shapes. CTPN [6], seglink [7], and CENet [8] regard the text region as a component sequence comprised of multiple character components, and utilize the region proposal method to detect text component regions and connect them by post-processing. Text detection methods based on region suggestion have to design anchors of various sizes, proportions, and skewness, and often suffer from the issue that it is difficult for the designed anchors to match text regions. Consequently, for indoor scene text with small target texts, the existing approaches for anchor design have difficulty of keeping a balance between the recall of small target texts and the computational cost, and may lead to an extreme imbalance between positive samples of small target texts and that of large target texts, which makes the involved models focus more on the detection of large target texts and thus neglect the detection of small target texts.

Text detection based on image segmentation exploits deep convolutional networks and up-sampling to fuse features to segment an image and determine whether each pixel of the image belongs to a text region. The accuracy and efficiency of these approaches are generally higher than those of region-suggestion-based scene text detection methods. EAST [9] and DDR [10] use multi-scale fusion features to directly predict the geometric information of text boxes where each pixel is located, which is an improvement on the time-consuming detection of text boxes using indirect regression of candidate boxes by region-based suggestion methods, but their detection accuracy needs to be improved. According to the basic idea of semantic segmentation, STD [11], PixelLink [12] and PSENet [13] adopt the global features of the image to predict the multi-task classification result of each pixel. DBNet [14] utilizes adaptive threshold graph for training to improve the segmentation results, and proposes Differentiable Binarization (DB) to tackle the gradient non-differentiable problem brought about by the training threshold map, thereby simplifying post-processing steps and improving inference speed. Compared with most text detection methods, DBNet achieves better detection accuracy and rate on multiple datasets. However, DBNet employs the feature pyramid structure FPN [15] for multi-scale information fusion in the feature fusion stage, and the simple FPN structure lacks consideration of feature space distance information and feature location information. This leads to insufficient fusion of semantic information of high-level features and detailed information of low-level features, and insufficient enhancement of global features. These deficiencies may cause possible issues such as missed or wrong detection of small target texts in complex scenes.

III. Text Detection Model

A. Network architecture

The proposed attention mechanism-based text detection network consists of three modules, which are feature extraction module, feature fusion module, and prediction output module. The overall network architecture is shown in Fig. 2.

The residual network [16] solves the issue that an excessively deep network can cause gradient explosion and disappearance, thus enabling the extraction of deeper features. Then, it can be conducive to the detection of small target texts. In this paper, we choose Resnet18 as the lightweight backbone network of the model in the feature extraction module, taking into account the issues of computational efficiency and feature detail information loss. Due to varying sizes of text’s length and width in indoor scenes, deformable convolution is added to all 3×3 convolutional layers of Conv3, Conv4, and Conv5 of the Resnet18 network. The deformable convolution can provide flexible perceptual fields and facilitate the detection of text with different aspect ratios.

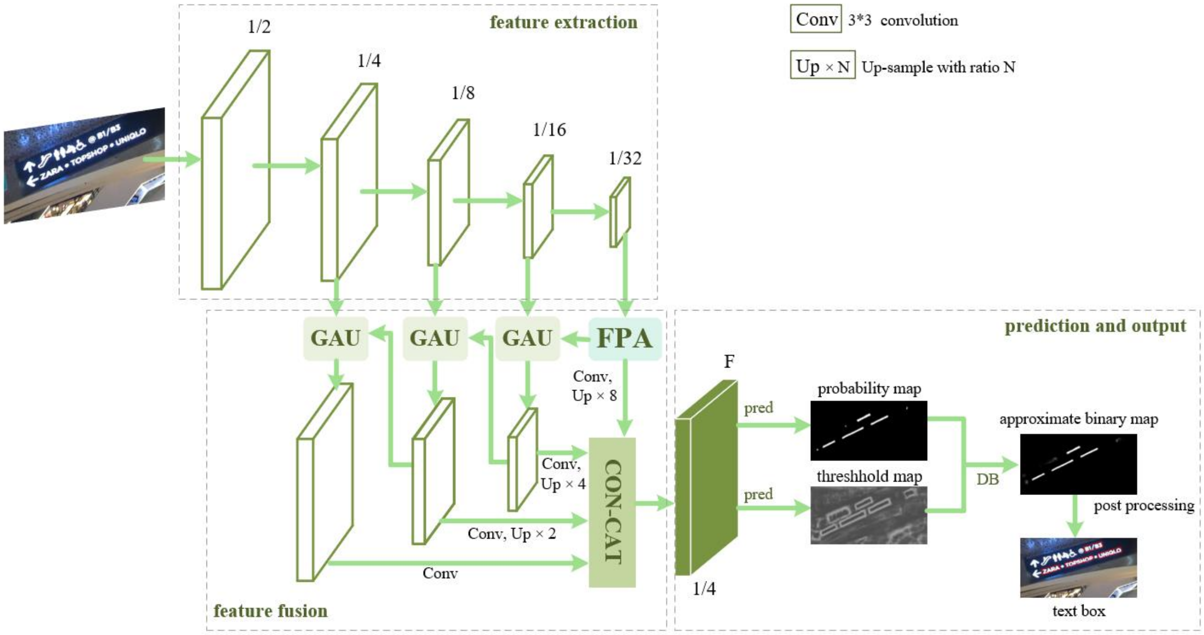


Figure 2. Network architecture of text detection method based on attention mechanism.

Even though Resnet18 achieves a high detection rate, its shallow network structure and small perceptual field lead to insufficient feature extraction and tend to miss medium-length texts, which greatly affect the detection of small target medium-length text. To address these two issues, we augment an attention mechanism to the feature fusion module of the model to recover the location details of high-level features from pixel level and to obtain a larger perceptual field. First, the last output layer of Resnet is connected to the Feature Pyramid Attention (FPA) module. FPA utilizes a three-layer convolution to fuse context information at different scales for high-level features and provides accurate pixel-level attention. Next, the Global Attention Upsample (GAU) module is introduced to deal with the remaining low-level features, and GAU takes the global context information as a guide for the computation of low-level feature mainly through the global average pooling layer. Finally, the obtained four feature maps are up-sampled to the same scale, and the resultant feature maps are cascaded to achieve the feature maps with fully fused information.

In the prediction and output module of the model, the fused feature maps are predicted and the probability map and threshold map are output. The values of each pixel point in the probability map and threshold map represent the probability of that pixel point being a target and the adaptive threshold of that pixel point, respectively. The approximate binary map is computed by combining the probability map and threshold map through the DB method proposed in [14]. The text box is generated by post-processing the probability map or the approximate binary map.

B. *Attention mechanism*

The principle of the attention mechanism can be summarized as focusing attention on some key information while ignoring secondary information. In indoor scene text detection tasks, the target text information is usually appearing in a complex context. The channel attention of SENet [17] and EncNet [18] does not consider pixel-level localization information. The grid artifacts caused by null convolution in Deeplab [19] and the global pooling operations of PSPNet [20] both result in the loss of pixel-level localization information. An image segmentation model with pixel-level attention is proposed in [21], which is able to provide pixel-level attention and recover pixel-level localization details. This paper borrows the idea of the pixel-level attention mechanism and incorporates it into the pyramid structure of the feature fusion module in the proposed model as a way of enhancing the location information of features and improving the performance of indoor text detection.

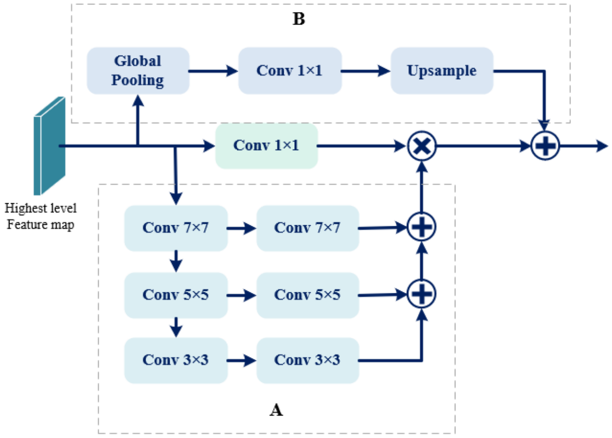


Figure 3. Feature Pyramid Attention Module (FPA).

1) Feature Pyramid Attention: The input of the Feature Pyramid Attention (FPA) module is the highest-level feature of output from the feature extraction network. As shown in Fig. 3, FPA possesses two branches, A and B, where branch A adopts a pyramid structure to fuse multi-scale context information. Since the resolution of high-level features is small, using larger convolutional kernels does not bring too much computational burden. So three sizes of convolutional kernels, 3×3, 5×5, and 7×7, are used in the pyramid structure. First, high-level features are put through three convolutional layers to extract multi-scale information, and then the attained information is passed through another three convolutional layers having the same kernel sizes as the previous ones and upsampled layer by layer to gain the features with fusion attention. Subsequently, the output features are multiplied pixel by pixel with the high-level features obtained from 1×1 convolution to achieve the weight enhanced features with attention. Considering the influence of global features, branch B introduces a global pooling layer to equalize the output of branch A, which avoids the loss of location information caused by using global pooling alone and further improves the performance of FPA module.

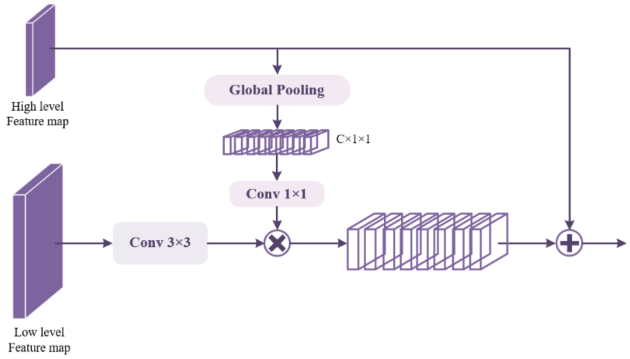


Figure 4. Global attention on sampling module (GAU).

2) Global Attention Upsample: Multiple features can be obtained after feature extraction using deep networks. The features obtained by passing through deeper networks are usually referred to as high-level features and otherwise as low-level features. High-level features contain rich semantic information and have strong classification ability, which can guide the localization of low-level features in the feature fusion process. At the same time, low-level features have higher resolution, which can help high-level features to recover details and pinpoint targets. In this paper, we exploit the GAU shown in Fig. 4 to attain global context information from high-level features through global pooling, and then utilize this information to weight and guide the low-level features. The implementation of the GAU module can be divided into three steps. First, the context information is obtained from the input high-level features through a global pooling layer followed by a 1×1 convolution layer, whereas a 3×3 convolution is applied to the input low-level features to reduce their number of channels. Next, the separately processed high-level features low-level features are multiplied to attain the weighted low-level features. Finally, the high-level features are up-sampled and then summed with the weighted low-level features to achieve the fused new features, which fully fuse the useful information of both the high-level and low-level features.

In summary, we introduce the FPA module and the GAU module to the feature fusion module in the model. FPA provides pixel-level attention information by extracting enhanced high-level features and enlarging perceptual field through the pyramid structure. GAU is applied in the decoder stage, and this module employs high-level features to weight low-level features and take the result as a guide to progressively recover pixel-level location details on low-level features. The attention-based feature fusion module fuses and enhances the input image features to gain features with rather more discriminative power than the input features, which helps the model locate the exact position of the target of interest more accurately and thus make better predictions.

C. DB module and loss function

1) Differentiable binarization: Traditional binarization methods transform a probability map into a binary map by setting a fixed threshold *t*. Subsequently, pixel-level segmentation results are grouped using heuristic techniques (e.g., pixel clustering), which in turn separate the text targets from the background.

$$B_{i,j} = \begin{cases} 1 & if\ P_{i,j} >= t, \\ 0 & otherwise. \end{cases} \tag{1}$$

As shown in Formula (1), when the pixel point $P_{i,j}$ is a text target with a probability value greater than or equal to a given threshold, the value of $B_{i,j}$ is 1 and it is identified as the target region, and otherwise as the background region. The fixed threshold value in the traditional binarization method is not differentiable and cannot be put into the network for learning. To tackle this issue, the differential binarization (DB) module adopts the feature map to additionally generate a threshold map *T* and introduces the DB method, as shown in Formula (2).

$$\hat{B}_{i,j} = \frac{1}{1+e^{-k(P_{i,j}-T_{i,j})}} \tag{2}$$

where $P_{i,j}$ and $T_{i,j}$ denote the probability map and threshold map predicted from the fused feature maps, respectively, $\hat{B}_{i,j}$ is the approximate binary map, and *k* is the magnification factor, which is set to 50. The curve of differentiable binarization function is similar to that of the traditional binarization. However, the former can be differentiated whereas the latter cannot. Hence, the function can not only facilitate the implementation of binarization, but also be put into the network for training and optimization, which improves the robustness of the model.

The improvement of detection performance brought by differentiable binarization can be explained in terms of back propagation. First, let: $P_{i,j} - T_{i,j} = x$, the binarization method can be expressed as $f(x) = \frac{1}{1+e^{-kx}}$. The loss of positive samples $loss_+$ and the loss of negative samples $loss_-$ can be expressed as:

$$\begin{aligned} loss_+ &= -log\frac{1}{1+e^{-kx}} \\ loss_- &= -log(1-\frac{1}{1+e^{-kx}}) \end{aligned} \tag{3}$$

The partial derivatives of the above two equations can be obtained as:

$$\frac{\partial loss_+}{\partial x} = -kf(x)e^{-kx}$$

$$\frac{\partial loss_-}{\partial x} = kf(x) \tag{4}$$

From (3) and (4), it is known that:

- The gradient is amplified by the coefficient k, accelerating convergence.
- The gradient has a significant amplification effect on the wrong prediction, which in turn promotes model optimization.

2) Text box generation for label: In training, the same labels are used for the probability map and the binary map. To generate the labels, the original text region G is shrunk by using the Vatti clipping algorithm [22] to gain the labeled region G_s, with the shrinkage offset D as shown in Formula (5).

$$D = \frac{A(1-r^2)}{L} \tag{5}$$

where A is the area of the original text area, L is the perimeter of the original text area, and the shrinkage factor r is empirically set to $r = 0.4$.

The labeling of the threshold map is different from that of the probability map and binary map, but their generation processes are similar. The original text region G is expanded to obtain the region G_d, and the expansion offset is calculated in the same way as the shrinkage offset D. The gap between G_d and the shrunken region G_s is used as the boundary of the text region. The shortest distance from each pixel point in the boundary to the four edges of the original text region G is calculated and then normalized to attain the label of the threshold map.

In the inference stage, the binary graph is obtained by binarizing the probability graph using a fixed threshold of 0.3 and the connected regions of the text is achieved accordingly. Using the offset D' to zoom in on the connected region can get the textbox, where D' is calculated as follows:

$$D' = \frac{A' \times r'}{L'} \tag{6}$$

where A' is the area of the coupling area, L' is the perimeter of the shrinkage linkage area, and the shrinkage rate r' is set to 1.5 empirically.

3) Loss function: The loss function is the weighted sum of loss of the probability map L_p, loss of the binary diagram L_b and loss of the threshold diagram L_t, as is shown in Formula (7).

$$L = \alpha \times L_p + \beta \times L_b + \gamma \times L_t \tag{7}$$

where α, β and γ are the three hyperparameters that control the loss balance and are set to 1, 1, and 10, respectively, depending on the relationship between the losses L_p and L_b. both L_p and L_b adopt binary cross entropy loss (BCE Loss) and apply OHEM [23] to difficult sample mining in calculating the loss so as to overcome the imbalance between positive and negative numbers. Therefore, L_p and L_b can be expressed as Formula (8).

$$L_p = L_b = \sum_{i \in S_l} y_i log x_i + (1 - y_i) log(1 - x_i) \tag{8}$$

where S_l is the sample set with a positive to negative sample ratio of 1:3. The threshold map loss L_t is the distance between the predicted result and the label in the expansion region G_d, which is calculated as shown in Formula (9).

$$Lt = \sum_{i \in R_d} |y_i^* - x_i^*| \tag{9}$$

where R_d are all the pixels in the expansion area G_d, and y_i^* is the label of the threshold map.

IV. Experiment

A. Dataset

Figure 5. Selected images of the dataset ICDAR2015.

The publicly available ICDAR2015 is chosen as the dataset, which consists of 1000 training images and 500 test images. The vast majority of the images are real indoor scenes captured by Google Glass with a resolution of 720 ×1280. Text language is English, and text instances are marked by quadrilateral boxes at the word level. The images in this dataset were captured without consideration of proximity positioning and image quality, with a relatively small percentage of text areas in the images, complex backgrounds, and poor text clarity.

B. Evaluation indicators

Three evaluation metrics, recall R (Recall), accuracy P (Precision), and average precision F (F-measure), are employed to evaluate the performance of the proposed model, which are calculated as follows:

$$Recall = \frac{\sum_{i=1}^{|G|} Match(G_i, D)}{|G|}$$

$$Precision = \frac{\sum_{i=1}^{|D|} Match(D_i, G)}{|D|}$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

We directly utilize the intersection ratio *IoU* of the truth box to the detection box to calculate Recall and Precision, where G is the set of truth boxes, D is the set of detection boxes, and $Match(X_i, Y)$ is the function for judging whether the two text boxes match and the degree of matching is measured by the *IoU* of the two text boxes. The threshold *t* is set to 0.5, and when the value of *IoU* is greater than the threshold *t*, the matching process succeeds and the function value is 1, and otherwise the process fails and the function value is 0.

C. *Experiment design*

In the experiments, we use Python 3.8 as the programming language together with PyTorch of version 1.9.1, and the hardware environment is composed of an NVIDIA GeForce GTX1080 Ti GPU and 11 GB video memory.

We adopt the Resnet18 network pre-trained by ImageNet as the backbone network for the model and do not use other datasets for pre-training. The number of iterations is set to 1200 and the training batch size is set to 16. The SGD optimizer is employed in all training processes and the momentum is set to 0.9. In addition, we adopt the poly learning rate strategy to set the learning rate of iterations: $Lr = lr \times (\frac{1-iter}{max\_iter})^{power}$. The initial learning rate lr is set to 0.007, $power$ is set to 0.9, and max_iter indicates the maximum number of iterations.

The training image data is enhanced by geometric transformation, and the enhancement methods include random rotation, random flip, and random cropping of the images, where the rotation angle range is set to the range $(-10°\sim10°)$. During the training process, each processed image is resized to 640× 640 to improve the training efficiency. During the inference process, an input image is set with the shortest edge while maintaining the aspect ratio.

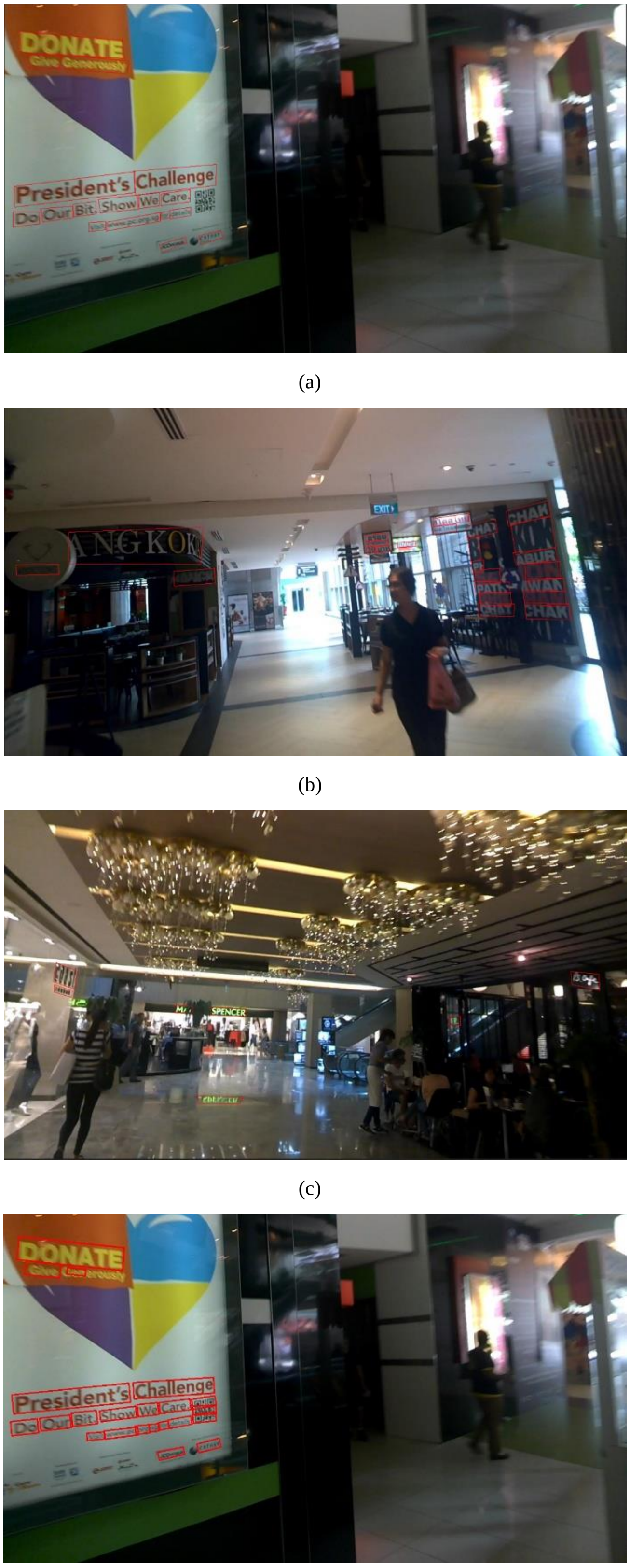
D. *Experimental results and analysis*

In order to verify the effectiveness of the proposed method for text detection in indoor scenes, experiments are conducted on ICDAR2015, a dataset with mainly indoor scene text images. The models selected for comparison are CPTN, SegLink, EAST, and DBnet, since they are representative and commonly used networks for text detection. In the inference process, the short edges of the test images are adjusted to 736, and simultaneously the aspect ratio is preserved, i.e., being the same as the original images. Considering the actual experimental environment, we do not pre-training the models 100k iterations using the dataset SynthText as in [14]. The experimental results of the proposed model and the selected models for comparison on the dataset are shown in Table 1.

TABLE I. Experimental results on dataset ICDAR2015

Indicators **Models**	**Precision (%)**	**Recall (%)**	**F-measure (%)**
CTPN[6]	74.1	51.9	61.0
SegLink[7]	72.8	76.9	74.8
EAST[9]	80.5	72.9	76.5
DBnet[14]	86.9	75.6	80.9
Ours	**87.3**	**77.7**	**82.2**

From the experimental results on the dataset ICDAR2015, it can be seen that the attention mechanism-based text detection model proposed in the paper significantly outperforms the region proposal-based text detection models in [6] and [7] and the segmentation-based text detection model in [9]. Compared with DBnet in [14] under the same condition, the accuracy, recall and average precision of the proposed model are promoted by 0.4%, 2.1%, and 1.3%, respectively. The experimental results demonstrate that the performance of the proposed model is better than other four text detection models in detecting text in indoor scenes, which suggests that augmenting the pyramid attention mechanism to the feature fusion stage can improve the text detection performance of the involved network to some extent.

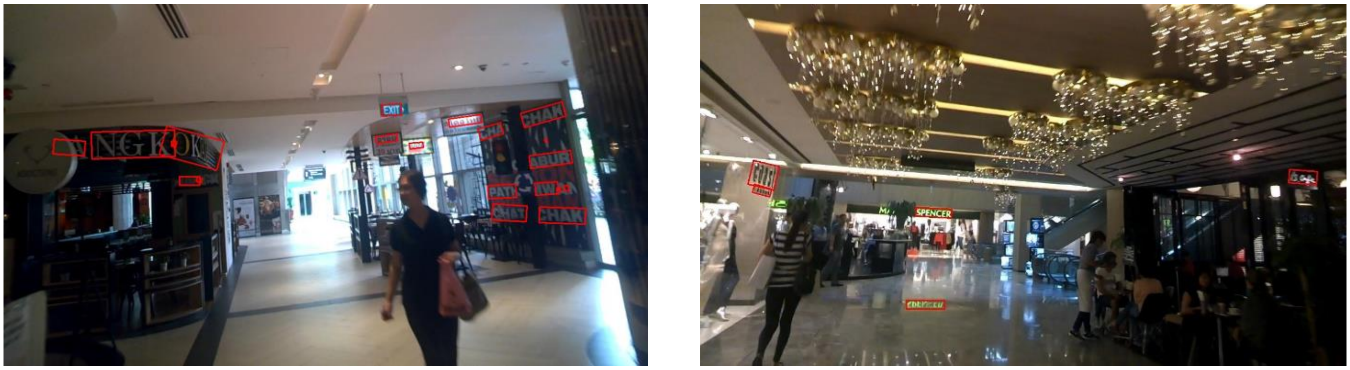


(a)

(b)

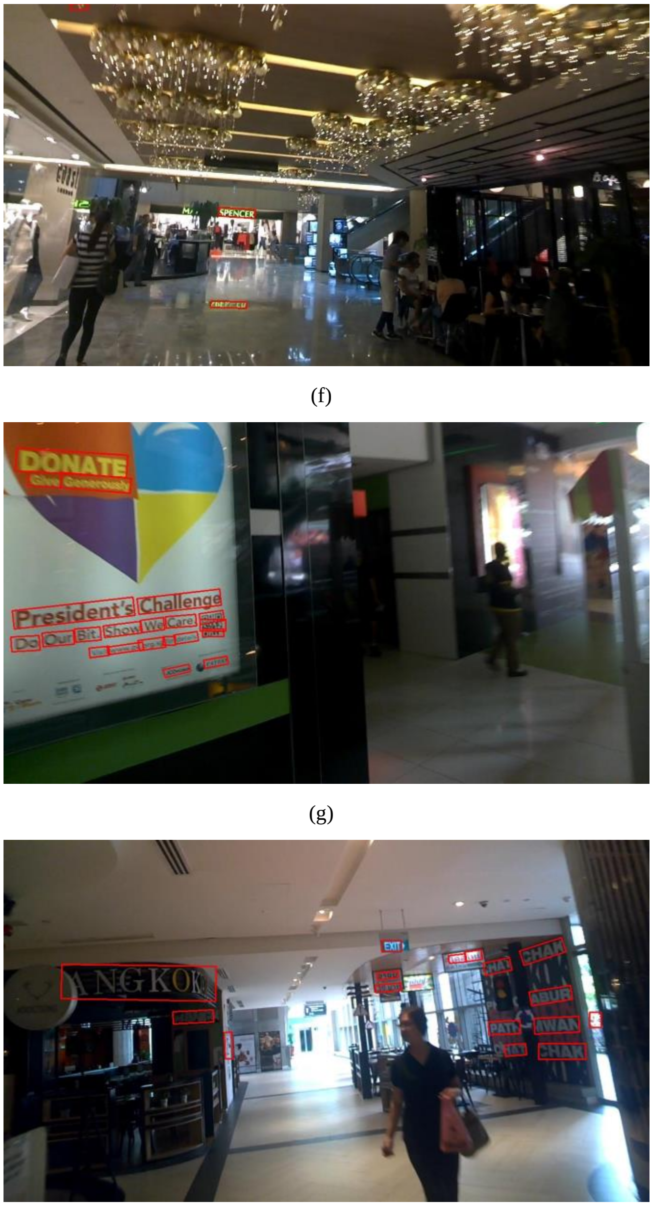
(c)

(d)



(e)

(i)



(f)

(g)

(h)

Figure 6. Experimental results on the indoor scene text dataset.

Moreover, to intuitively illustrate the distinction between the proposed model and DBnet, a typical example of the images processed by them in the above experiments is depicted in Fig. 6, where (a)-(c) represent the ground-truth, and (d)-(f) and (g)-(i) denote the predicted results of DBnet and the proposed model, respectively. It can be shown that some texts are missed or mistakenly detected in (d)-(f), whereas they are correctly found in (g)-(i). This implies that adding the attention mechanism to promote the localization function from pixel level, can reduce missed or falsely detected cases.

V. Conclusion

In this paper, we propose an attention-based text detection model for indoor scenes with complex backgrounds and small detection targets. The model utilizes Resnet18 with deformable convolution as the feature extraction network, adds the Feature Pyramid Attention module and the Global Attention Upsample module to the feature fusion stage, employs the Differentiable Binarization method to obtain the approximate binary map required for training. The model can reduce the interference of complex backgrounds and enhance the detection capability of small target texts through the enhancement of pixel-level location information of high-level features and the fusion of context information. The experimental results on the dataset ICDAR2015 show that the model performs well on the task of indoor scene text detection, and also that the designed attention mechanism improves the detection performance of the model. In future work, we shall further optimize the model and apply it to other types of text detection datasets.

References

[1] X. Gao, M. Mo, H. Wang, and J. Leng, “Recent advances in small object detection,” Journal of Data Acquisition and Processing, 36(03), pp.391-417, 2021.

[2] A. Gupta, A. Vedaldi, and A. Zisserman, “Synthetic data for text localisation in natural images,” In: Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, pp.2315-2324, 2016.

[3] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, “TextBoxes: A fast text detector with a single deep neural network,” In: Proceedings of the AAAI Conference on Artificial Intelligence, pp.4161-4167, 2016.

[4] Y. Zhu, and J. Du, “Sliding line point regression for shape robust scene text detection,” In: Proceedings of the International Conference on Pattern Recognition. 2018, pp.3735-3740.

[5] L. Deng, Y. Gong, Y. Lin, J. Shuai, X. Tu, Y. Zhang, M. Zheng, and M. Xie, “Detecting multi-oriented text with corner-based region proposals,” Neurocomputing, 334, pp.134-142, 2019.

[6] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, “Detecting text in natural image with connectionist text proposal network,” In: Proceedings of the European Conference on Computer Vision, pp.56-72, 2016.

[7] B. Shi, X. Bai, and S.J. Belongie, “Detecting oriented text in natural images by linking segments,” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.3482-3490, 2017.

[8] J. Liu, C. Zhang, Y. Sun, J. Han and, E. Ding, “Detecting text in the wild with deep character embedding network,” In: Proceedings of the Asian Conference on Computer Vision, pp.501-517, 2018.

[9] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, “EAST: An efficient and accurate scene text detector,” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.2642-2651, 2017.

[10] W. He, X. Zhang, F. Yin, and C. Liu, “Deep direct regression for multi-oriented scene text detection,” In: Proceedings of the IEEE International Conference on Computer Vision, pp.745-753, 2017.

[11] Y. Wu, and P. Natarajan, “Self-organized text detection with minimal post-processing via border learning,” In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5010-5019, 2017.

[12] D. Deng, H. Liu, D. Cai, and X. Li, “PixelLink: Detecting scene text via instance segmentation,” In: Proceedings of the AAAI Conference on Artificial Intelligence, pp.6773-6780, 2018.

[13] X. Li, W. Wang, W. Hou, R. Liu, T. Lu, and J. Yang, “Shape robust text detection with progressive scale expansion network,” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.9328-9337, 2019.

[14] M. Liao, Z. Wan, C. Yao, K. Chen and, X. Bai, “Real-time scene text detection with differentiable binarization,”In: Proceedings of the AAAI Conference on Artificial Intelligence, pp.11474-11481, 2020.

[15] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.936-944, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.770-778, 2016.

[17] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.7132-7141, 2018.

[18] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, “Context encoding for semantic segmentation,” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.7151-7160, 2018.

[19] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” arXiv preprint arXiv:1706.05587, 2017.

[20] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.6230-6239, 2016.

[21] H. Li, P. Xiong, J. An, and L. Wang, “Pyramid attention network for semantic segmentation,” arXiv preprint arXiv:1805.10180, 2018.

[22] Vati. Bala R, “A generic solution to polygon clipping,” Communications of the ACM 35(7), pp.56-64, 1992.

[23] A. Shrivastava, A. Gupta, R. Girshick, “Training region-based object detectors with online hard example mining,” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.761-769, 2016.

Hybrid BERT for Dialogue State Tracking

Yuhong He
Department of Computer and Information Science
Southwest University
Chongqing, China
yuhonghe592@gmail.com

Yan Tang
Department of Computer and Information Science
Southwest University
Chongqing, China
ytang@swu.edu.cn

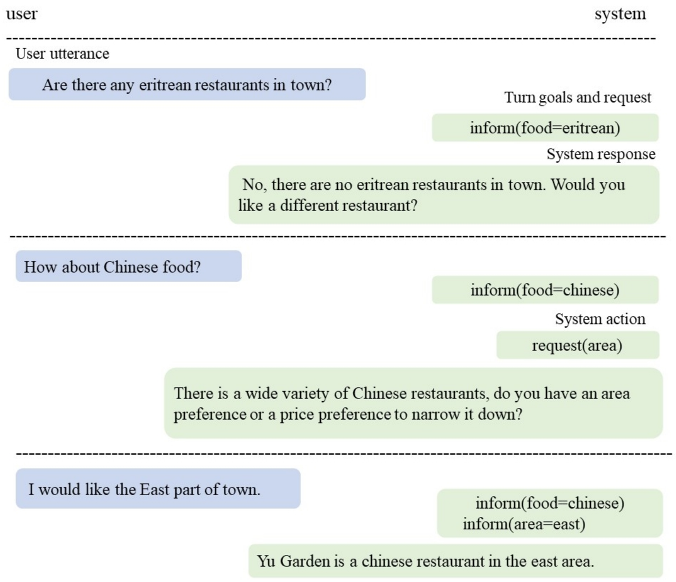
***Abstract*—In a task-oriented dialog system, the goal of dialogue state tracking (DST) is to predict the current state given all previous dialogue contexts. Recently, many deep learning-based methods have been proposed for the task. However, existing complex models cannot achieve good results on small-scale annotated datasets due to the lack of training data, making the model difficult to train and converge. This paper proposes a Hybrid Bert model (HyBERT) which uses the pre-trained model BERT to encode utterances and candidate slot-value pairs separately, then leverage semantic similarities between the representation of the utterances and the candidates to compute the belief state distribution. In addition, we use focal loss instead of traditional cross-entropy to solve the problem of imbalance between positive and negative samples. Experimental results demonstrate that HyBERT outperforms all previous methods, achieving new state-of-the-art results on the standard WoZ 2.0 dataset.**

***Index Terms*—Dialog System, State Tracking, BERT, Focal Loss, WoZ 2.0**

I. Introduction

As an important branch of natural language processing, a task-oriented dialogue system aims to gradually collect relevant information used to achieve specific dialogue goals during multiple rounds of dialogue with users and ultimately helps users obtain specific information services [1]. It generally includes four parts: natural language understanding (NLU) for obtaining the user's intention, dialogue state tracking (DST) to update the dialogue state, dialogue policy (DP) for making corresponding decisions, and natural language generation (NLG) for generating responses. To prevent errors from propagating between modules, NLU is usually integrated into the DST for joint modeling training. From the entire process of the task-based dialogue system, it can be seen that the dialogue state tracking task is one of its core components. Its main task is to use the dialogue information between the system and the user in multiple rounds to update the dialogue status in real-time. The state of the dialogue determines the response method adopted by the system and is the basis for the system to make decisions. The dialog state is expressed in the form of a tuple (slot, value), and all possible slot-value pairs are given in a predefined domain ontology. Figure 1 shows an example dialogue with annotated turn states.

With the renaissance of deep learning, many methods directly learn the dialogue state tracking task end-to-end [2]–[4] and achieve competitive performance on standard DST datasets such as DSTC 2.0 [5] or WoZ 2.0 [6]. However, most of these methods still have some limitations.



Are there any eritrean restaurants in town?

inform(food=eritrean)

No, there are no eritrean restaurants in town. Would you like a different restaurant?

How about Chinese food?

inform(food=chinese)

request(area)

There is a wide variety of Chinese restaurants, do you have an area preference or a price preference to narrow it down?

I would like the East part of town.

inform(food=chinese)
inform(area=east)

Yu Garden is a chinese restaurant in the east area.

Fig. 1. An example dialogue with annotated turn states.

Many approaches require training a separate model for each slot type in the domain ontology [2], [7]. Therefore, the number of parameters is proportional to the number of slot types, making the scalability of these models become a significant issue. On the other hand, deep learning relies much large-scale labeled data, and insufficient labeled data is difficult to meet the needs of deep learning model training leading to the performance of the dialogue state tracking model based on deep learning on small-scale annotated datasets is often not ideal.

The ultra-large-scale pre-trained models represented by BERT [8] and GPT [9] make up for the shortcomings of insufficient annotation data, and greatly improve the performance of the dialogue state tracking model. The existing pretrained-based state tracking models simply contact utterances and slot-value pairs together as the input of BERT, then performs classification through a fully connected layer [10], [11]. It does not fully consider the relationship between the dialogue text and the slot type. Since all slot-value pairs in the ontology

DOI reference number: 10.18293/DMSVIVA22-007

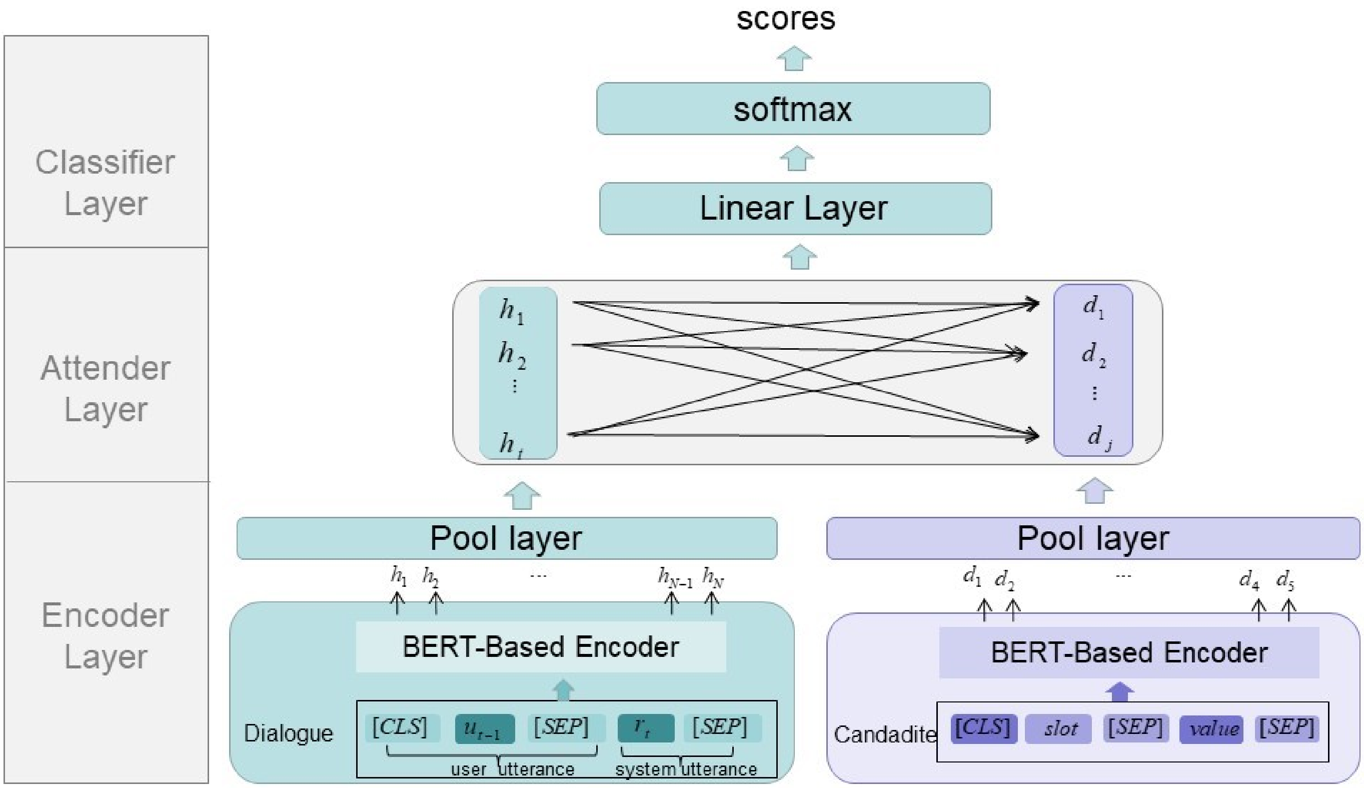


Fig. 2. The architecture of HyBERT, consists of two encoders, an attender and a classifier.

domain are repeatedly encoded, a lot of time is spent in the encoding phase.

In this paper, we proposed a hybrid Bert model (HyBERT) which uses two BERTs to encode user-system utterance and all slot types in the ontology, then calculates the similarity of the state of the last hidden layer of the two BERTs, performs classification to determine whether the currently encoded slot-values appears in this round of dialogue type. Since all slot-value pairs in the ontology are encoded only once at this stage, the speed of overall encoding is much faster. Otherwise, considering that there are only a few slot types in a conversation, but various combinations of slot-values in the definition ontology, so a large number of negative samples will be generated. To solve this problem, focal loss [12] is used for training. The contributions of this work are summarized as follows:

- HyBERT employs BERT [8] to learn and utilize better representations of not only the current utterance but also the previous utterances in the dialogue.
- HyBERT no longer models each slot value pair separately, the number of model parameters does not increase with the ontology size which greatly enhances the model scalability.
- To solve the problem of the imbalance between positive and negative samples, HyBERT uses focal loss instead of cross-entropy to calculate the loss.
- Experimental results demonstrate that HyBERT achieves new state-of-the-art results on the standard WoZ 2.0 [6] dataset.

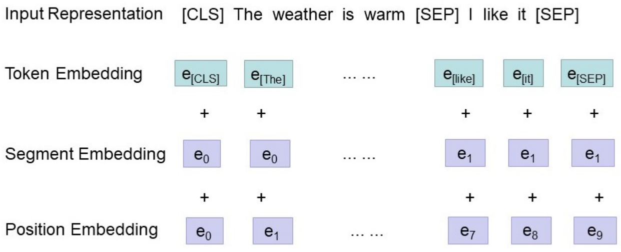


Fig. 3. BERT input representation

II. RELATED WORK

Previous works mainly focus on encoding the dialogue context, and slot-values with neural networks such as CNN, RNN (or variants of RNN like LSTM and GRU) and attention mechanisms for DST. StateNet [13] applies the Long Short-Term Memory (LSTM) [14] to track the inner dialogue states among the dialogue turns. GLAD [2] is comprised of an

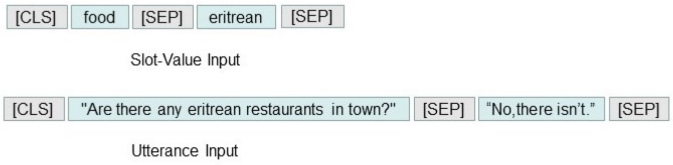


Fig. 4. The input of two encoders

encoder module that uses a global-locally self-attentive to encode all utterances and a scoring module to consider the contribution from the user utterance and previous system actions. GCE [4] is based on GLAD using only one recurrent network with global conditioning. Belief Tracking [15] employed 7 independent bi-directional LSTMs to encode the user and system utterances.

Recently, several pre-trained language models, such as ELMO [16], GPT [9] and BERT [8], were used to achieve state-of-the-art results on many NLP tasks. COMER [17] is a BERT-based hierarchical encoder-decoder model, that generates state sequence based on user utterance. BERT-DST [18] uses BERT as a dialogue context encoder and makes parameter sharing across slots. SimpleTOD [10] is a simple approach to task-oriented dialogue that uses a single, causal language model trained on all sub-tasks recast as a single sequence prediction problem.

III. OUR METHOD

As shown in Figure 2, HyBERT consists of two encoders, an attender and a classifier. The encoder includes an utterance encoder and a candidate slot-value encoder which the former transforms the user and system utterances in the dialogue into a sequence of utterance embeddings and the latter encodes each slot-value pair defined in the domain ontology, both of them use BERT-based pretrained model. The attender leverage semantic similarities between the utterances representation and the slot-value vectors in the candidate set for making a prediction. Finally, the classifier decides whether the current candidate was expressed in the user utterance.

A. Encoder

The encoder layer includes utterance and candidate encoder, both of which employ BERT to construct a sequence of embeddings. BERT is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right contexts in all layers. The basic model structure of BERT consists of multiple layers of Transformer, including two pre-training tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP).

The mask language model uses a method similar to cloze filling, masking part of the words in the input text and restoring them to the original words through BERT to avoid information leakage problems caused by the two-direction language model. 15% of the Word Pieces subwords of the input sequence in Bert are masked, and the masked position is replaced with a [MASK] mark with a probability of 80%, replaced with other words in the vocabulary with a probability of 10%, and also remained the same as the original word with a probability of 10%. Next sentence prediction needs to determine whether the input B is the next sentence of the input A, to construct the relationship between the two sentences.

The beginning of the sequence is always indicated by a special token [CLS], and the two input texts are separated by [SEP] which also represents the end of the sequence. The input representation of BERT is composed of the sum of token embedding, segment embedding and position embedding, shown in Figure 3. BERT uses Word pieces to segment the word, and then converts the word into a vector representation through the word vector matrix; the block vector is used to encode which piece the current token belongs to; the position vector is used to mark the absolute position of the token. The output is a sequence of vectors, one for each input token.

The utterance and candidate encoder input representation are shown in Figure 4. The input of the utterance encoder consists of the system utterance and the user utterance from the current turn t, denoted as $x_t = [CLS, u_{t-1}, SEP, r_t, SEP] \in R^N$ which N is the length of the sequence, and [CLS] is a special classification token always at the start of every input sequence, [SEP] token is used to separate the two input texts, also mark the end of the sequence. Another candidate encoder takes the descriptions of slot-value pairs as input, denoted as $y_i = [CLS, s_i, SEP, v_i, SEP] \in R^5$. Then simply passed to BERT to get the output vectors:

$$\begin{aligned} h_t &= BERT\left(x_t\right) \\ d_i &= BERT\left(y_i\right) \end{aligned} \tag{1}$$

$h_t = [h_{1t}, \ldots, h_{Nt}] \in R^{N \times d_{emb}}$ and $d_i = [d_{1i}, \ldots, d_{5i}] \in R^{5 \times d_{emb}}$ is the outputs of two encoder, is the embedding size, referred to as utterance embedding and candidate embedding, with one embedding for each token.

B. Attender

The attender takes the sequence of utterance embedding and the set of candidate embedding as input and calculates similarities between two representations. In this way, information from the utterances and the candidate are fused. In order to keep the dimensions consistent to facilitate calculations, we first pool the output of the two encoders, (i.e., the output of the last hidden layer of the BERT) to obtain two 768-dimensional vectors $\overline{h_t}$ and $\overline{d_i}$. Given the t-th turn of utterance embedding $\overline{h_t}$ and i-th candidate embedding $\overline{d_i}$, attention $A_{t,i}$ calculates the similarity as follows:

$$\operatorname{Atth}\left(h_t \cdot d_i\right) = \frac{\overline{h_t} \cdot \overline{d_i}}{\sqrt{d_{emb}}} \tag{2}$$

Dividing by the vector dimension d_{emb} is to avoid the result of the dot product being too large due to the vector dimension.

C. Classifier

Based on the output vector of the attender, the probability of the candidate slot-value pair being relevant is :

$$o_t = \operatorname{sofmax}\left(W_{t,i} + b\right) \in R^2 \tag{3}$$

where the transformation matrix W and the bias term b are model parameters. $softmax$ is an activate function, that squashes the score to a probability between [0,1]. To enhance the generalization ability of the model, dropout and layer normalization are used after this module. At each turn, the classifier is used to determine whether the encoded i-th slot-value pair exists in the current dialogue text, then add the predicted state to the state list, otherwise, it will be ignored.

Considering that only a few slot-value pairs are mentioned in each turn of dialogue, there will be a large number of negative samples when generating slot-value pairs according to the predefined domain ontology. Training these large numbers of negative samples with positive samples will make the model hard to learn in rare classes. Therefore, we choose focal loss [12] in the model to replace the traditional cross-entropy to calculate the loss for the training model.

Focal loss aims to solve the class imbalance problem by reshaping the standard cross-entropy loss such that it down-weighs the loss assigned to well-classified examples. Combined with the two classifications of our work, the standard cross-entropy loss formula is as follows:

$$CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases} \tag{4}$$

For notational convenience, define p_t :

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{5}$$

and rewrite $CE(p, y) = CE\left(p_t\right) = -\log\left(p_t\right)$.

Introduce a weighting factor $\alpha \in [0, 1]$ to balance the importance of positive and negative examples.

$$CE\left(p_t\right) = -\alpha_t \log\left(p_t\right) \tag{6}$$

Focal Loss introduces another tunable focusing parameter $\gamma \geq 0$, adding a modulating factor $(1 - p_t)^\gamma$ to the cross-entropy loss, to reshape the loss function to down-weight easy examples and thus focus training on hard negatives.

$$FL\left(p_t\right) = -\left(1 - p_t\right)^\gamma \log\left(p_t\right) \tag{7}$$

Finally, focal loss formulate defined as follows:

$$FL\left(p_t\right) = -\alpha_t\left(1 - p_t\right)^\gamma \log\left(p_t\right) \tag{8}$$

IV. EXPERIMENT

A. Dataset

To evaluate the effectiveness of our proposed approach, we use the standard WoZ 2.0 [6] dataset. The dataset consists of user conversations with dialog systems designed to assist users in finding a restaurant in the Cambridge, UK area. The ontology gives details of all possible dialogue states and includes a list of attributes termed requestable slots that users can use to constrain the search, such as the food type or phone number. It also provides a list of informable slots that the user can ask a value for once a restaurant has been offered. Each informable slot has a set of possible values. The dataset contains three requestable slots ("price range", "food", "area") and six informable slots ("address", "phone", "postcode", "price range", "food", "area"). A total of 1200 turns of user and system dialogue texts are included, with divided training, validation and test sets according to the ratio of 3:1:2.

TABLE I
COMPARATIVE EXPERIMENTAL RESULTS

Method	**Evaluation Indicators**	
	Joint Goal	**Turn Request**
Neural Belief tracker	**84.2%**	**91.6%**
GLAD	**88.1%**	**97.1%**
StateNet	**88.9%**	-
COMER	**88.6%**	-
BERT-based tracker	**90.5%**	**97.6%**
Seq2Seq-DU	**91.2%**	-
HyBERT	**91.4%**	**97.7%**

B. Training Details

The model is trained end-to-end using the Adam optimizer with a batch size of 16 and sets the warmup proportion to 0.1. We selected the best fine-tuning learning rate 2E-5 on the training set and choose the base version of BERT that consists of 12 Transformer layers [19], each with a hidden size of 768 units and 12 self-attention heads. The dropout ratio of the classifier module is set to 0.25.

C. Result

Two evaluation metrics, turn request accuracy and joint goal accuracy, are used to evaluate the performance on single domain DST. Turn request accuracy calculates the percentage of the correct prediction of requestable slots for each round and the joint goal accuracy is the accumulation of informable slots accuracy.

We make a comparison with the following existing models: Neural belief tracker [6], GLAD [2], StateNet [13], COMER [17], BERT-based tracker [11] and seq2seq-DU [20].

- Neural belief tracker [6]: Conduct reasoning on pretrained word vectors, and combine them into representations of user utterance and dialogue context.
- GLAD [2]: Model uses self-attentive RNNs to learn a global tracker that shares parameters among slots and a local tracker that tracks each slot, and computes semantic similarity with predefined ontology terms.
- StateNet [13]: It is independent of the number of values, shares parameters across all slots, and uses pre-trained word vectors instead of explicit semantic dictionaries.
- COMER [17]: BERT-based hierarchical encoder-decoder model, generates state sequence based on user utterance.

TABLE II
THE IMPACT OF DIFFERENT COMPUTATIONAL ATTENTION METHODS ON MODEL ACCURACY.

Method	Evaluation Indicators	
	Turn Request	Joint Goal
Perceptron	**96.96%**	**64.09%**
Bilinear	**97.32%**	**88.21%**
Scaled dot	**97.75%**	**91.43%**

TABLE III
ACCURACIES OF HYBERT AND HYBERT-W/O FL ON WOZ2.0 DATASET.

Method	dropout=0.2		dropout=0.25	
	Turn Request	Joint Goal	Turn Request	Joint Goal
HyBert-w/o FL	**96.38%**	**89.00%**	**96.44%**	**78.73%**
HyBERT	**97.63%**	**90.58%**	**97.75%**	**91.43%**

- BERT-based tracker [11]: A unified sequence-to-sequence model based on BERT, given a dialog context and a candidate slot-value pair, the model outputs a score indicating the relevance of the candidate.
- Seq2Seq-DU [20]: It formalizes schema-guided DST as a sequence-to-sequence problem using BERT and pointer generation.

As shown in Table I, HyBERT achieves the highest performance, 91.4% on joint goal accuracy and 97.7% on turn request accuracy on WOZ2.0 [6]. Like the BERT-based tracker and Seq2Seq-DU, the proposed methods both use the BERT pretraining model to fine-tune the dialog state tracking task. Different from their work, we use focal loss when calculating the loss and only encode all candidate terms on the ontology once.

V. ABLATION STUDY

We conduct ablation studies on HyBERT with experience the influence of different ways of calculating attention in the attender on the model, and the importance of focal loss to the accuracy of the entire model. The results indicate that all the components of HyBERT are indispensable.

To investigate the effectiveness of different ways of calculating attention in the attender, we try several mainstream calculation methods such as multilayer perceptron, bilinear, and scaled dot product used in the model respectively. The formula for each method of calculating similarity is as follows:

$$\operatorname{Attn}(h_t, d_i) = \begin{cases} h_t^T W d_i & \text{bilinear} \\ \frac{h_t^T d_i}{\sqrt{d_{emb}}} & \text{scaled} \\ w^T \tanh\left(W\left[h_t; d_i\right]\right) & \text{perceptron} \end{cases} \tag{9}$$

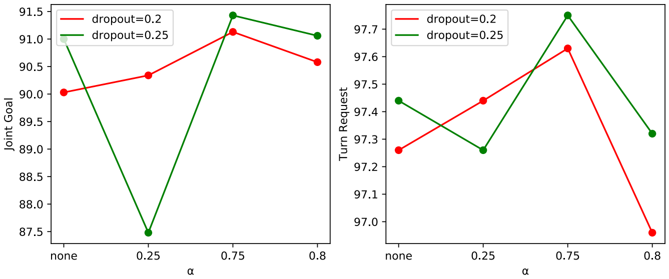


Fig. 5. The effect of hyperparameters in focal loss

As shown in Table II, we can see that the scaled dot product which we have chosen to use achieves the highest performance. We also explore the influence of hyperparameters in focal loss on model accuracy. Fixed $\gamma = 2$, results for various α are shown in Fig.5. Since there is an extreme imbalance between positive and negative samples during training, the model can often achieve better performance when α is larger. It can be seen that when $\alpha = 0.75$, joint goal accuracy and turn request can reach their peaks, but still increasing the value of will harm the performance of the model.

Table III shows the effect of the presence of focal loss on the model. The γ and α were respectively set to 2 and 0.75 in the experiment. One can observe that the accuracy of the training model using traditional cross-entropy decreases significantly which indicates that using the focal loss can focus training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training.

VI. CONCLUSION

In this paper, we propose a novel dialogue state tracker, referred to as HyBERT, which employs BERT to encode utterances and candidate descriptions respectively and shares all of its parameters across the slots in the predefined ontology, achieves state-of-the-art joint goal accuracy, and turn request accuracy on the WOZ2.0 dataset. Experiment results also demonstrate the effectiveness of focal loss use. In future work, we will experiment on more large-scale datasets such as the MultiWOZ and SGD.

REFERENCES

[1] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

[2] Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*, 2018.

[3] Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016.

[4] Elnaz Nouri and Ehsan Hosseini-Asl. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*, 2018.

[5] Matthew Henderson, Blaise Thomson, and Jason D. Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A., June 2014. Association for Computational Linguistics.

[6] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

[7] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*, 2019.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[10] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*, 2020.

[11] Tuan Manh Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8034–8038. IEEE, 2020.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[13] Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. Towards universal dialogue state tracking. *arXiv preprint arXiv:1810.09587*, 2018.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Osman Ramadan, Paweł Budzianowski, and Milica Gašić. Large-scale multi-domain belief tracking with knowledge sharing. *arXiv preprint arXiv:1807.06517*, 2018.

[16] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[17] Liliang Ren. *Scalable and accurate dialogue state tracking via hierarchical sequence generation*. University of California, San Diego, 2020.

[18] Guan-Lin Chao and Ian Lane. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *arXiv preprint arXiv:1907.03040*, 2019.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[20] Yue Feng, Yang Wang, and Hang Li. A sequence-to-sequence approach to dialogue state tracking. *arXiv preprint arXiv:2011.09553*, 2020.