Application of Slow Intelligence Framework for Smart Pet Care System Design

Shi-Kuo Chang[1], Wen-Hui Chen[2], Wen-Chyi Lin[3] and Christopher Lee Thomas[1]
[1]Department of Computer Science, University of Pittsburgh, USA ({schang, clt29}@pitt.edu)
[2]Graduate Institute of Automation Technology, National Taipei University of Technology, Taiwan (whchen@ntut.edu.tw)
[3]Department of Electrical and Computer Engineering, University of Pittsburgh, USA (wel69@pitt.edu)

Abstract—**This article presents the design of a smart pet care system based on the slow intelligence framework for providing pets with suitable living conditions that closely mirror their natural habitat. By integrating heterogeneous information from various sensing data, the smart environment-aware pet care system can adaptively adjust the setting of temperature and humidity that best fits for the pet through iterative slow intelligence computation. Simulations of two case studies were provided to illustrate the application of the proposed system for pets such as snakes and dogs. The simulation results demonstrate the feasibility of the proposed approach to the design of smart pet care systems.**

Keywords- Smart pet care systems; slow intelligence systems; environment-aware software engineering

I. Introduction

Unlike human health care systems, pet care systems require more autonomous mechanisms to perceive and respond to the changes of environmental conditions as pets are unable to alert their caretakers if an anomalous condition arises. As pets have their specific living conditions to thrive, a pet care system for one species may not suitable for another. Therefore, component reuse software architecture is beneficial for the design of a pet care system.

Software reusability is essential and plays an important role in building a scalable system from software component reuse and integration. In this study, we proposed a component-based software framework based upon slow intelligence systems to integrate existing software components into self-regulating and adaptive systems. A slow intelligence system (SIS) is a general-purpose system characterized by being able to improve its performance over time [1]. An SIS is characterized by employing super-components, in the sense that multiple components can be activated either sequentially or in parallel to search for solutions. We have developed a visual specification approach using dual visual representations, and the user interface to design a component-based system based upon the dual visual representations [2].

The proposed software framework emphasizes the design of reusable software components so that the developed system can be easily applied to various pets. To implement an environment-aware system for pet care, an adaptive control strategy that adopted the principles of slow intelligence systems was proposed. The proposed environment-aware system can sense and react on the environment accordingly to provide pets with a proper temperature and humidity setting through continuous interaction with the environment by evolutionary computation that carries out adaptive control operations.

The caring system for a pet snake is different from the one for a pet dog or a pet cat. To validate the proposed system is appropriate for various pets, we used the dog care and the snake care as two illustrative examples to demonstrate the feasibility of the proposed framework for the application of a smart pet care system.

The remainder of this article is organized as follows. Section 2 introduces the framework of the proposed pet care system. Section 3 describes the application of the proposed framework for monitoring pet snakes and pet dogs. Section 4 provides simulation results and discussions. Conclusions are drawn in Section 5.

II. The Framework of SIS-based Pet Care System

Considering the software usability and scalability, we proposed a component-based software architecture for pet care system design. The proposed framework, as shown in Fig. 1, consists of eight software components, including Universal Interface, SIS Server, System Interface, Alerter, Pet Activities Monitor, Environment Sensor, Image Sensor, and Image Processor.

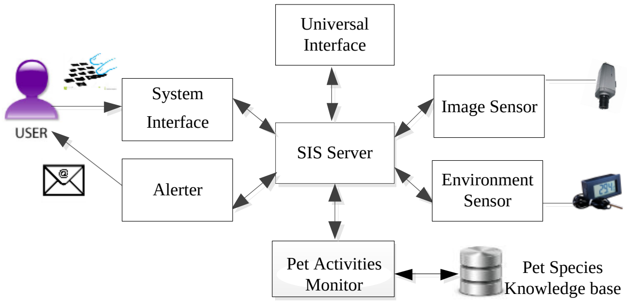


Fig. 1. The software architecture of the proposed pet care system.

2.1 Slow Intelligence System (SIS) Server

The SIS server is responsible for routing and processing messages among components. When the SIS server accepts a request message from a component, the message will be processed and routed to the designated components by the SIS server as responses. All request and response messages are encoded in the XML format for usability across the Internet. Some particular messages were defined for the SIS server to execute specific operations. For example, messages Create, Kill, and Activate are used to enable the SIS server to create, kill and activate a user defined component, respectively.

2.2 System Interface (SI)

The System Interface component provides a graphic user interface (GUI) for users to specify system properties, including pet names, which e-mail address an alert should be sent to, the thresholds used by Image Processor, and the parameters required for reasoning algorithms. Note that this component does not receive any messages from the SIS server, and has no control logic inside it. It is only a GUI for users to specify system properties. The properties set by this component are stored in a shared database for other components to access. Thus, there is no need to design extra communication between this component and other components. If the user does not set the properties, default properties will be used.

2.3 Universal Interface (UI)

Universal Interface is used to simulate routing messages among components. It is useful when testing the developed system by observing routing messages. The user can enter a message on the left panel, and observe the corresponding output message on the right panel. Any message sent from one component to another component will be displayed on the right panel.

2.4 Pet Activities Monitor (PAM)

Pet Activities Monitor is designed to recognize pet activities by analyzing sensor data from Environment Sensor and Image Sensor. Different types of sensors require different data processing. PAM consists of four units: a data preparation unit, an image processing unit, an image recognition unit, and a decision unit. The data preparation unit is to convert sensing data to feature vectors. The image processing unit is to process images captured from Image Sensor and perform feature extraction for the image recognition unit, while the image recognition unit is to recognize pet activities. The decision unit is to make a decision of whether an alert needs to be sent to the caretaker about an anomalous condition.

When an MSG 33 message is received by PAM, the image processing unit will start to load the image captured from Image Sensor and its timestamp described in MSG 33, and perform specific activity recognition algorithms.

2.5 Alerter

The Alerter component is responsible for receiving an MSG 38 message and sending an alert e-mail or text message to the user to notify an anomaly is detected so that the user can take action accordingly.

2.6 Environment Sensors (ES)

The Environment Sensors component is designed to read and store all available sensor data, such as temperatures and humidity for observing environmental conditions. In addition to analog sensor readings, Environment Sensor also can handle binary sensor data such as the status of limit switches.

2.7 Image Sensor (IS)

The Image Sensor component is designed to read and store the image frames captured from the external camera. The location where the camera is installed depends on the purpose of applications. In the case of pet snake care, the camera could be mounted atop the water bowl, while in the case of dog care, the camera could be placed in a location where the dog can be easily observed. When a new captured image is generated, a message MSG 33 will be generated and sent to notify the SIS server. The stored images are analyzed by the image processing unit in Pet Activities Monitor. As the observed pet may not sit near the server or may not even be in the same building as the place where the pet lives, it is necessary for Image Sensor to support different camera inputs, such as IP cams and webcams for remote access.

III. Applications of the Proposed Framework for Pet Care Systems

3.1 Description of Pet Snake Care

Snakes are poikilothermic animals and require environmental heat for various bodily activities. Although commercially available snake Vivaria can provide a housing enclosure with heating, water bowl, and covering, most of them lack of sensing and adaptive temperature control mechanisms for snake habitat monitoring, which can lead to a severe injury to pet snakes when the heater malfunctions.

The ideal habitat for pet snakes is not only monitoring the temperature on the hot side or the under tank heater to avoid a thermal burn, but also on the cool side to prevent the temperature from dropping too low. In addition, the humidity of the environment must also be monitored and kept in a proper setting. Even with these constraints, pet snakes still need the temperature lowered or raised according to their activities at all times, such as during shedding, after eating, etc.

The activity of a pet snake reveals some information about its status. For example, if it goes into the water bowl, it usually indicates the snake either wants to cool off or it feels sick. Although the information of living environment can be obtained by acquiring data from environment sensors, it is insufficient to adjust the temperature and humidity properly merely according to the sensor readings. As the proper setting may vary depending on their activities. In order to observe basic activities of a pet snake, an IP camera atop the water bowl is used.

A pet snake needs a suitable enclosure and some essential equipment inside the enclosure, like under tank heaters, water bowls and environmental sensors such as temperature and humidity sensors. The size and material of the enclosure may vary from one snake species to another. So does the equipment inside the enclosure. For example, some small snakes become anxious when living in a big space. The selection of the hardware equipment for pet snakes needs domain experts and beyond the scope of this research. Therefore, we assume the hardware enclosure as well as the required equipment have already properly decided and installed, and only focus on the design of the software system for a smart pet care system.

3.2 Pet Activities Monitor for Snake Care

The Pet Activities Monitor component for pet snake care contains the fuzzy inference engine and fuzzy rule base for making a decision under uncertainty. The fuzzy decision starts with sensor data reading from environment sensors and goes through the fuzzification process. The inference engine works with attribute values that have fuzzy memberships defined, and produces a fuzzy output using max-min of inference. The output values are then de-fuzzified to a crisp value by the use of centroid de-fuzzification. The fuzzy rules and the parameters of membership functions are specified by users through the SI component.

Image Sensor provide the information of whether or not the observed snake is drinking out of the water bowl or lays in the water bowl by analyzing the captured image through vision-based scene analysis. In this experiment, we used the speeded up robust features (SURF) algorithm as an example for snake head detection to detect whether the head or the body of the snake is present in the water bowl and specified a value between 0 and 1 for these two possibilities.

The training images were collected and manually annotated for SURF feature descriptors. The SURF descriptors are feature vectors describing certain sections of an image. The image processing unit first computes the SURF feature descriptors over all the training images for both drinking and soaking. Fig. 2 shows two sample images with SURF feature descriptors for a Python Regius drinking and soaking in a water bowl. This is done to avoid re-computation of these values for each image received. The rationale is that images whose SURF descriptors match given thresholds are likely to be showing the same object.

When an MSG 33 message is received, the image processing unit will start to load the image and perform the SURF algorithm to detect whether the head or the body of the snake is present in the water bowl. If either of these are true, the probability of there being a problem is increased.

3.3 Iterative Slow Intelligence Computation

To determine an appropriate setting of temperature and humidity, we adopted iterative slow intelligence computation that contains four primitive phases, namely: enumeration, adaptation, elimination, and concentration defined in slow intelligence systems. A slow intelligence system is able to observe and act on the environment to achieve adaptive temperature and humidity control through the iterative computation process.

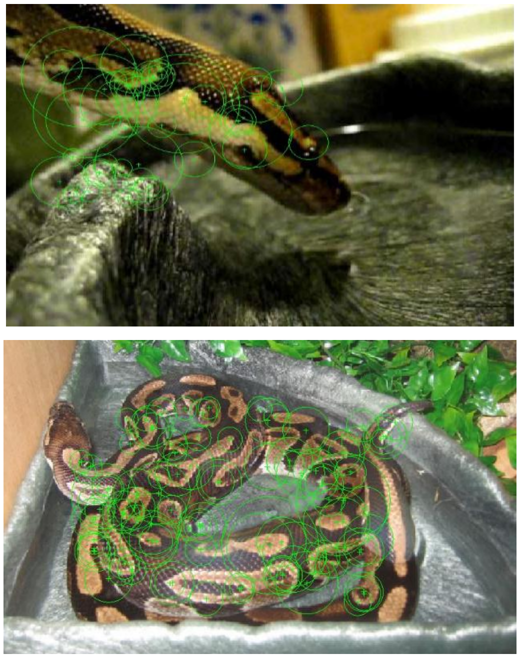


Fig. 2: Sample images with SURF feature descriptors for a Python Regius drinking and soaking in a water bowl.

In the proposed system, the possible combinations of temperature and humidity settings are considered as candidate control plans, which is the process of enumeration in slow intelligence system. At first, a set of fuzzy rules was built based on domain experts and was stored in the fuzzy rule base. Elimination is the process of ruling out un-matched SURF feature descriptors when comparing the query image with training images. This makes the recognition process more efficient as resources are only focused on matched descriptors, which is a formation of concentration. The proposed system is environment-aware by constantly interacting with the environment. The setting of the under tank heater could cause the changes of the temperature and humidity in the environment, so an update of the setting according to sensing data is required, which is the process of adaptation in slow intelligence computation. In Fig. 3, The SIS framework is shown where the circle and the timing control illustrates a super-component of an iterative slow intelligence system [3].

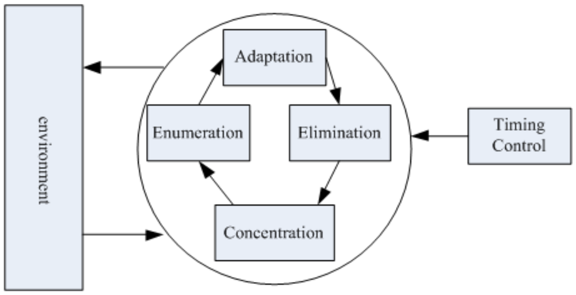


Fig. 3. The SIS framework where the circle and the timing control illustrates a super-component of an iterative slow intelligence system

The information obtained from Image Sensor is integrated with the information from Environment Sensor as data fusion so that the procedures performed by Pet Activities Monitor can decide if an alert should be sent or not. The procedures performed by PAM are described as follows.

Step 1: When a captured image is loaded, the SURF descriptors are computed on that image to identify regions of interest in the image.

Step 2: The SURF descriptors are matched using the *k*-nearest neighbor algorithm against the pre-computed descriptors of every training image in the training set. In this study, we set $k = 1$ to find the closest matching descriptor and then compute the distance. That is, each vector associated with each descriptor in the captured image will be matched with one vector of the set on each training image. The matched vector is the one that has the least Euclidean distance.

Step 3: For each of the matched sets, the overall normalized match strengths are computed. The cosine distance between each of the matched vectors was adopted as similarity measure.

Step 4: Those matches whose distance is greater than the experimentally determined threshold are considered as bad matches and are discarded. Note that this step is critical as it will eliminate many bad matches, but also preserve the good matches as the process of concentration. The image containing the most matches is used and the other image matches are discarded.

Step 5: The count of matches for the final image above the threshold value is input into a sigmoid function which translates the number of matches into a probability value to be used by the fuzzy inference engine inside the Pet Activities Monitor component.

3.4 Pet Activities Monitor for Dog Care

Living in an excessively cold or hot space can make a dog feel uncomfortable or even become life-threatening. Not all dogs are created equal. Different breeds of dogs have different hair coats. Breeds from cold climates usually have a downy coats and are much better at conserving heat than at cooling themselves. Therefore, the range of comfortable ambient temperatures for dogs to live varies from one breed to another. It is not the temperature but also the humidity that can affect dogs' health. If the humidity goes too high, dogs are unable to cool themselves, leading to their temperature raising to dangerous levels. As such, the best temperature and humidity settings in the dog housing environment should take dog breeds into consideration.

As the proposed system is designed in the framework of component reuse, only the PAM component in Fig. 1 needs to be modified for the design of a dog care system. Therefore, in this section, we only describe the dog breed recognition algorithm implemented in the PAM component. With this function, the proposed SIS-based pet care system can adaptively adjust the ambient temperature and humidity to a suitable setting for the pet dogs according to their breeds.

The knowledge required for proper temperature and humidity settings can be extracted and collected in the knowledge base through interviewing veterinarians and dog experts. The PAM component comprises of several distinct classifiers and is integrated with the SIS Server to identify a dog breed from the camera. The information flow for dog breed recognition in the pet care system is shown in Fig. 4.

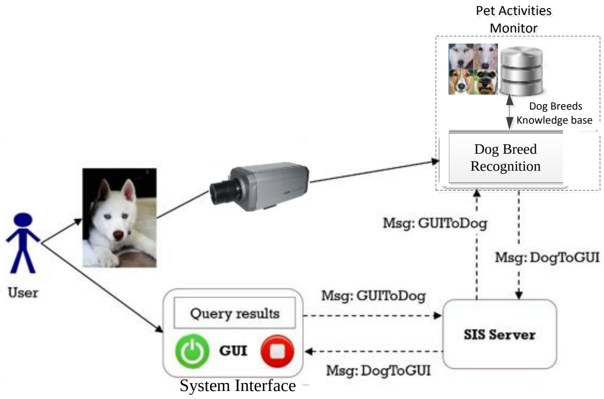
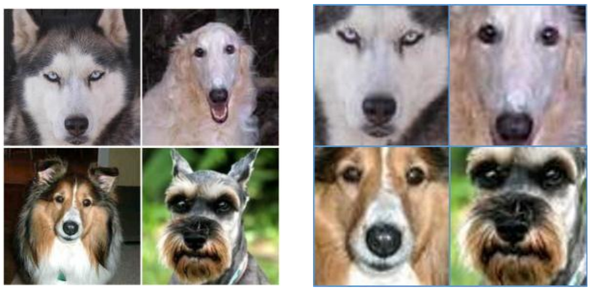


Fig. 4. The information flow of dog breed recognition in the pet care system.

The dog face detection and dog breed recognition algorithms are adapted from the Haar feature-based cascading classifier [4] and the FisherFace classifier, respectively. To process images rapidly and achieve high detection rates, the integral image is applied to compute the features used by the Haar feature detector. The Haar features are a set of directional filters that consist of different combination of rectangles. Based on the integral image, different sizes of Haar feature-based filters can be used to detect the region of interest (ROI) rapidly from the background in an image.

The sample dog images are adopted from the Stanford Dogs dataset [5] which contains images of 120 dog breeds. Fig. 5 shows four sample images and their cropped normalized images to a size of 70×70 pixels. This normalization is used to avoid extremely weights and maintain numerical stability.



(a) sample images (b) normalized images

Fig. 5: Sample images from the Stanford Dogs dataset

Fig. 6 shows that the eyes and nose of a Borzoi dog are detected by two Haar feature filters and thus this sub-image can be accepted for further processing.

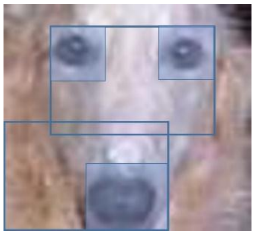


Fig. 6: Eyes and nose of a Borzoi dog are detected by Haar feature filters

As a particular case of ensemble learning, we generated a cascading classifier by combining increasingly more complex classifiers to allow background areas of the image to be quickly discarded while saving computing power on more dog face-like regions. As illustrated in Fig. 7, in the first stage of cascade, we remove the most unwanted backgrounds by using a coarse grained filter to favor speed. The last stage of cascade is a fine grained filter which can handle more detailed object features, then the detected result is passed to the FisherFace classifier for performing dog breed recognition.

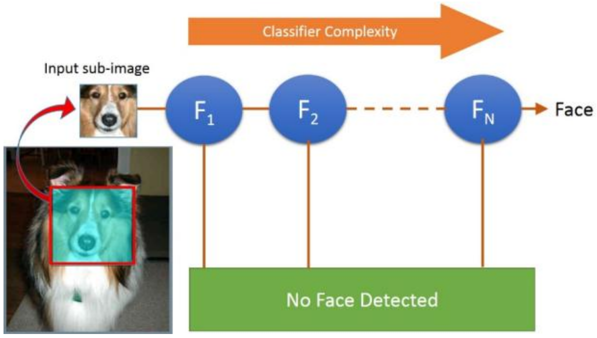


Fig. 7. Schematic diagram of the cascading classifier

Based on the Fisher's linear discriminant analysis, the FisherFace is proved to be robust against variation in lighting and facial expressions [6]. Fig. 8 shows three FisherFaces of the dog images in Fig. 5 (b).

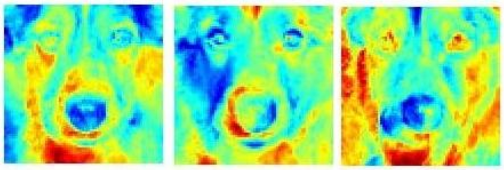


Fig. 8: Sample FisherFaces of dog images

IV. Simulation Results

In this section, we describe the simulation results of the proposed approach by using the SIS testbed developed by the University of Pittsburgh. This testbed is a platform designed for developing SIS projects, and is available online at http://people.cs.pitt.edu/~chang/163/interface/SequenceSIS.htm

4.1 Simulation for Pet Snake Care

Before activating the system, some information needs to be specified as system properties through the SI component. SI provides a graphic interface for users to determine pet names, alert e-mail address, the thresholds for Image Processor, and the parameters for the fuzzy inference engine.

The camera at the time of experiment was sitting on top of the snake enclosure pointed down towards the water bowl, and the Image Sensor component polls the camera every five seconds for a new image. Once the image is acquired from the camera, it is saved to a temporary location for further analysis. The message MSG 33 is broadcast to the SIS system to acknowledge that the image processing unit can begin processing the image.

The Environment Sensor component reads sensor data and broadcasts it to the system. In this experiment, we prepared a file with sensor readings as sensor data for simplicity. Environment Sensor repeatedly reads temperature data from the file to demonstrate the function of Environment Sensor. In the practical system, it would read from the real sensors.

The image processing unit begins processing images when it receives the message which contains the path to the captured image file from Image Sensor. The image has the keypoint descriptors calculated over it, which are then matched with the soaking and drinking data sets. After processing, it outputs two probabilities (in bowl probability and drinking probability) which are sent to Pet Activities Monitor.

The drinking probability and the in-bowl probability are obtained from the result of the image processing unit, and are used directly as membership values via one-to-one membership function mapping. The membership functions for the linguistic variables (warm, cold, hot) of temperature values and the centroid de-fuzzification are illustrated in Fig. 9.

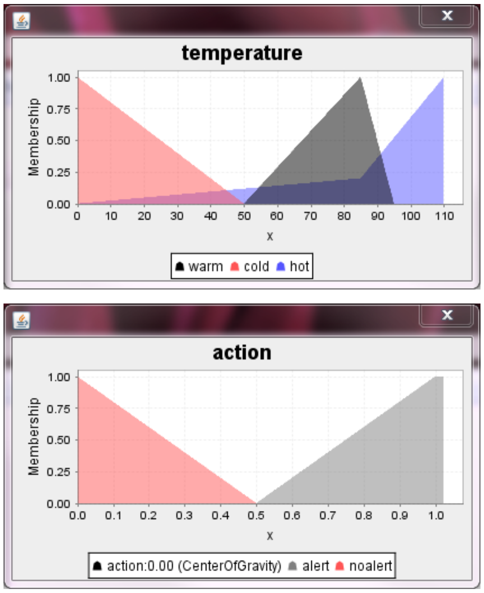


Fig. 9: The membership functions for evaluating temperature values and centroid de-fuzzification

Pet Activities Monitor will send an MSG 38 message if an anomalous condition exists. When an anomaly is detected, Alerter will send an alert to notify the pet owner, if no anomalous condition exists, the message MSG 38 will not be generated. Note that the fuzzy inference engine waits until it receives MSG 31 and MSG 34. To demonstrate the system, we set the temperature to 80 degrees Fahrenheit and took an image of a Python Regius drinking from the water bowl to represent the captured image from Image Sensor, as shown in Fig. 10. The drinking probability is obtained from Image Processor, and indicates the drinking probability is 0.6 which is higher than the in-bowl probability. Therefore, the system detects the snake is drinking from the bowl.

When Alerter receives the message MSG 38 from Pet Activities Monitor to indicate that the condition is anomalous enough to warrant the user's attention. When we check the e-mail inbox of the e-mail address specified in System Interface, we can find the email message describing that an anomaly is detected.

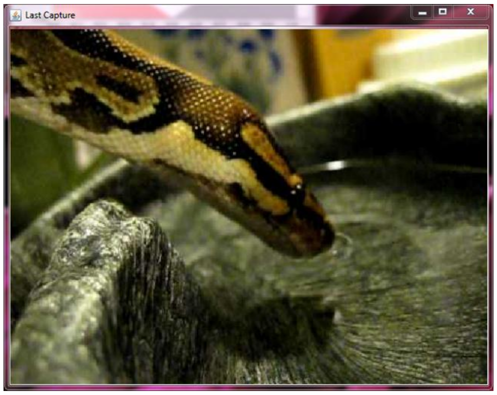


Fig. 10: The image used to represent the snake drinking from the water bowl.

4.2 Simulation for Pet Dog Care

Fig. 11 illustrates the information flow for the dog breed recognition. In this experiment, we use dog images as the pet dog in front of the webcam to demonstrate the effectiveness of the proposed system. Once the system recognizes the dog, the GUI will display the message contains the resulting dog breed less than 1/20 second. The system is working at 20 frames per second and can be adjusted according the hardware capability. The program is implemented as a joint effort of Java, OpenCV, Python, and C++ languages.

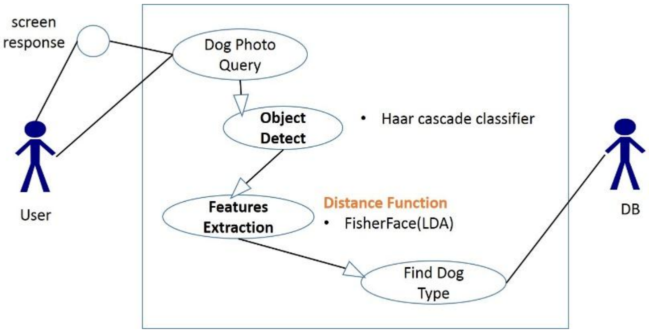


Fig. 11: The information flow for the dog breed recognition.

In this experiment, seven photos from four dog breeds, including the Eskimo, Shetland sheepdog, Borzoi, and Schnauzer are examined. A snapshot of the experimental results is shown in Fig. 12.

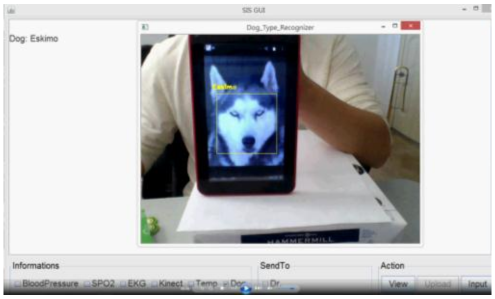

Dog: Eskimo

Fig. 12: A snapshot of the experimental results from a live demo.

V. Conclusions

In this study, a smart environment-aware pet care system has been established for providing pets with suitable living conditions by the fusion of heterogeneous sensors and the iterative slow intelligence computation. The proposed framework has component reuse and scalable features that make the proposed system can be easily extended and transferred to various pet care systems. The experimental test was performed using the SIS testbed and has shown that the proposed system could provide potential benefits for pet care.

While the system was currently developed in the simulation stage, it demonstrated the abilities to detect anomalous conditions and alert caretakers. The largest obstacle to generalizing the system is the specificity of the training image dataset required for the computer vision routines to work accurately. The results of this study provide a good case study in automatic caretaking which may have implications for autonomous healthcare systems for other animals. The scope of the system can be expanded by incorporating with spatially distributed sensors to monitor pet conditions and serve as a caregiver to manage the environmental conditions, watch for particular behaviors, feed pets and cooperatively pass messages through the network to specified locations.

References

[1] S. K. Chang, "A general framework for slow intelligence systems," *International Journal of Software Engineering and Knowledge Engineering*, vol. 20, pp. 1-15, 2010.

[2] S. K. Chang, Yao Sun and Yingze Wang, "Component-based Slow Intelligence System", Journal of Internet Technology http://jit.niu.edu.tw, January 2013.

[3] S. K. Chang, W. H. Chen, Bin Kao, L. Kuang, and Y. Z. Wang, "The design of pet care systems based upon slow intelligence principles," *Int'l Journal of Software Engineering and Knowledge Engineering*, 2014.

[4] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp 511-518, 2001.

[5] http://vision.stanford.edu/aditya86/ImageNetDogs/

[6] P. N Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," IEEE Trans. Pattern Analysis and Machine Intelligence, v.19 n.7, p.711-720, July 1997.