Journal of Computers Vol. 32 No. 1, 2021, pp. 95-101
doi:10.3966/199115992021023201008

Using Deep Learning to Track Stray Animals with Mobile Device

Rung-Ching Chen[*], Qiao-En Liu, Chung-Yen Liao

Department of Information Management Chaoyang University of Technology, Taichung 413, Taiwan, ROC
{crching, s10714604, s10514901}@cyut.edu.tw

Received 1 June 2019; Revised 8 November 2019; Accepted 6 January 2020

Abstract. The purpose of this paper is to protect stray animals and reduce the social problems they cause through mobile devices. We wrote an app on the mobile device to explicitly identify the dog or cat. The system combined with the backend repository and placed the Open Street Map (OSM) to show the geographic location of the animal. Through this app, users can know where the stray animals are, which type, and what kind of assistance is needed. At the same time, if the user finds that there are stray animals in his location, but the APP does not display. The user can identify the animals through the system upload the location, type, and other issues to the database. In the part of the observation, we used You Look Only Once (Yolo) to train and put the training module into the system for the identification of stray animals with a precision of 95%. The system can reduce the problems caused by stray animals to increase the willingness of people to adopt stray animals and let shelter reduce the cost of dealing with stray animals.

Keywords: deep learning, Yolo, Open Street Map

1 Introduction

Stray animals have become a problem that plagues humans, and the problem of euthanasia has attracted people's attention. After domestication and reproduction, there are many types and types of pets. After taming and breeding, types of pet species are various. Since pet stores and puppy mills are everywhere, people are easy to purchase pets for fashion or a sudden whim. Due to the difficulty of caring for or training a pet, people may give up their pets due to boredom, aging, illness, disability, and the like. The report from Environment & Animal Society of Taiwan, which points out the total number of 650,000 animals was euthanized in animal shelters between 1999 and 2008 in Taiwan. Roughly 73% of the total number of animals in shelters was euthanized. Moreover, the number of animal shelter euthanasia is increasing; it is estimated that approximately 100,000 animals were euthanized per year from 2006 [1]. The number of stray animals in public animal shelters are 113,025, 40.84% adoption rate, 40.41% euthanasia rate, 18.75% natural mortality rate in 2013 [2].

To track stray animals, we use Open Street Map to locate the habitat of stray animals. The OSM database contains records on all of the features found on OSM: from natural features such as roads, rivers, and hills to human-created ones such as parks, schools, hospitals, shops, and restaurants (see Fig. 1). Each data record in the database also has a variety of "tags" attached to it, which help describe what the record is related to [3].

We use deep learning to detect the type of animal through image recognition, in which we use YOLO. YOLO is refreshingly simple. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection [4-5]. We used the camera to shoot in our system and then used Yolo to identify stray animals. Our experiments are mainly to detect dogs and cats, and then together with the experimental process and system architecture will be mentioned in Section 3 Methodology.

[*] Corresponding Author

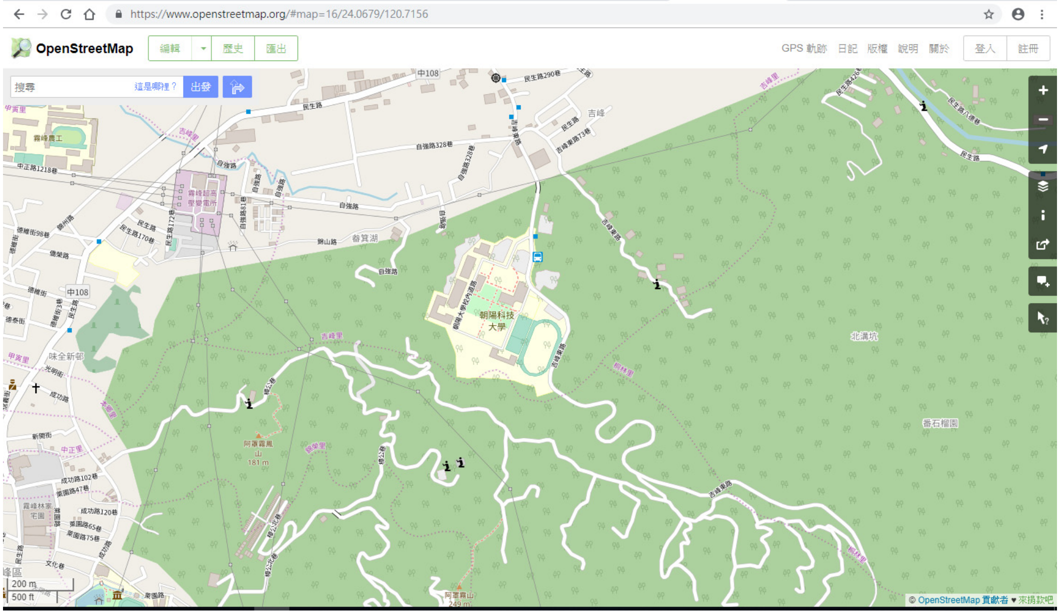

Fig. 1. OSM on the website (www.openstreetmap.org)

2 Related Works

We refer to six papers related to animal feedback. Pons [6] uses depth-based tracking in machine learning to detect animal behavior in combination with Kinect Sensor, which is a cat. First, the cat’s head, body, and tail are distinguished, and then the cat’s behavior is distinguished by centroid and vector. Finally, a Knowledge-based decision tree is established. Due to the limited range of sensor detection, it can only be indoors or in a relatively small open space.

Monteiro [7] places the sensor on the guide dog and detects the behavior of the guide dog and the object in front when assisting the visually impaired. They use AlexNet to integrate with GoogleNet and then use the support vector machine to classify the behavior of the guide dog into a different condition than the object in front. They are finally implemented on the system, according to various circumstances, voice feedback to the visually impaired.

Kolandaisamy [8] is an APP for stray animals. The experimental object is a dog. Socially present information about the owner and the dog, collect information about the dog’s need for rescue, and gather information about the dog’s disappearance. The system is a built-in waterfall mode and displayed on the website. There is no sensor and no learning.

Xu [9] uses uncrewed aerial vehicles (UAV) to track the use of wild animals on wireless sensor networks (WSNs). UAV does not only need to identify wild animals from a distance, but also automatically plan the movement path. Kuo [1] uses inquiry-based learning to simulate problems caused by stray dogs in Taiwan on NetLogo. They showed the number of stray dogs, habitat, ligation, euthanasia, and adoption on the simulator. Zhang [10] uses R-CNN and Single Shot Detector to identify cat breeds. And implement the identity system on the Android phone.

3 Method

We use Unity to build the system to Android and IOS. OSM is referenced utilizing the project UnitySlippyMap mentioned in Deng [11]. In conjunction with OSM, we use the GPS sensor of the phone. The open dataset of [12] is used in our paper. Before we train, we will first frame the objects we want to detect. They are then classified into two categories based on the data set: dogs and cats, then divided into 90% training and 10% testing. Then, the architecture of the YOLO model is shown in Fig. 2, which was extended from GoogleNet. The original Yolo built a 24-layer convolution layer followed by two layers of fully connected layers. Some layers use a 1 * 1 reduction layer to reduce the depth of the feature map. Finally, the shape of 7 * 7 * 30 tens is output from the last layer [4-5].

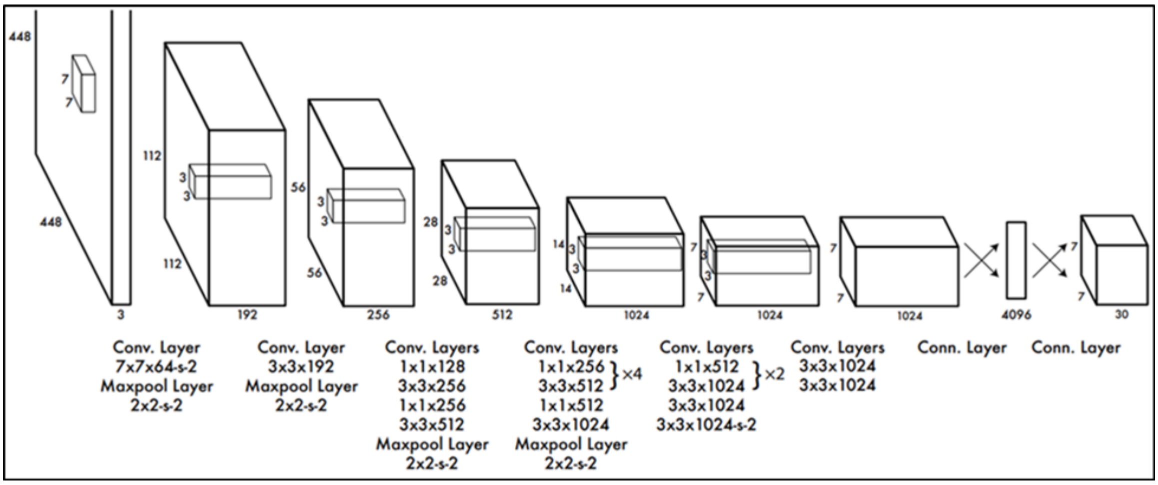

Fig. 2. Yolo/GoogleNet Architecture [3]

YOLO combines the initial scattered object detection steps into a single neural network. YOLO can predict each bounding box through the features of the entire image, and simultaneously calculate the probability of each bounding box for each class. YOLO is not only from the whole image. The detection of objects and end-to-end training and calculations, as well as real-time operations, still maintains high precision [4-5]. Fig. 3 shows the Yolo model is based on the regression problem. It is synchronized on the graph by dividing into even grids and predicts bounding boxes, confidence in those boxes, and class probabilities.

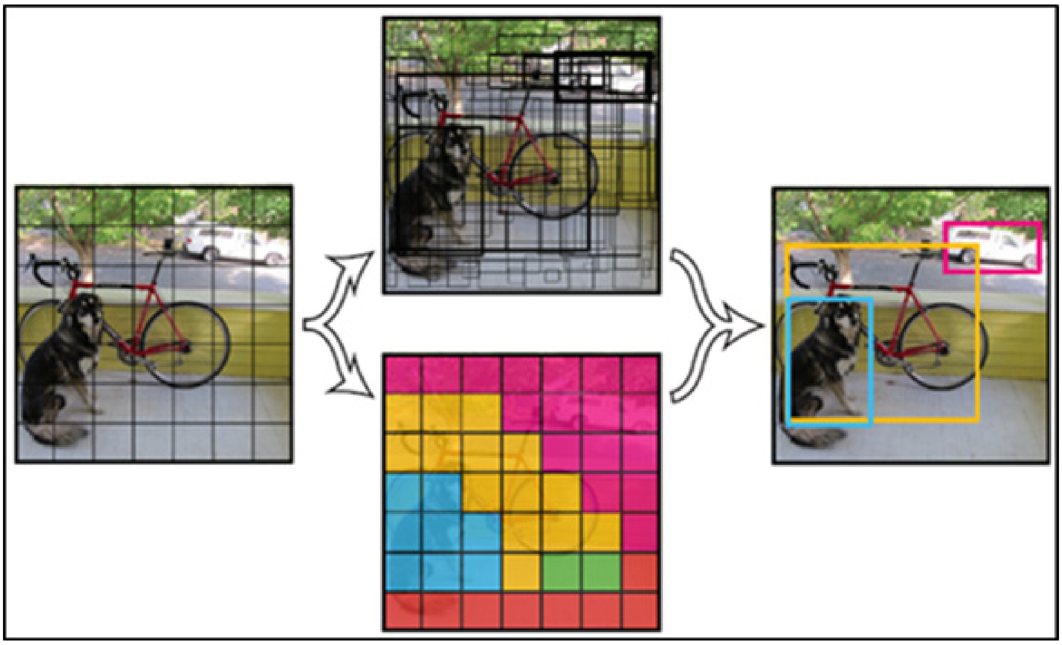

Fig. 3. A simplified illustration of the YOLO object identification process [3]

Our system will first enter the login interface. Users must first add new users if they are using it for the first time. After logging in, they will see an interactive interface with a map displayed by OSM with "tags," indicating the stray animals at the location. "Tags" have been added by other users in the past. However, these "tags" are only displayed on our app and will not be provided to OSM. Click on "tags" to display photos and information about stray animals. The information in these "tags" comes from our database. The method of adding "tags" and information about stray animals is to provide a photo-taking function through our system, as well as record positioning and animal-related information. If the animals are successfully identified, the user can be uploaded to the database. And add a "tags" to the OSM on the interactive interface (See Fig. 4).

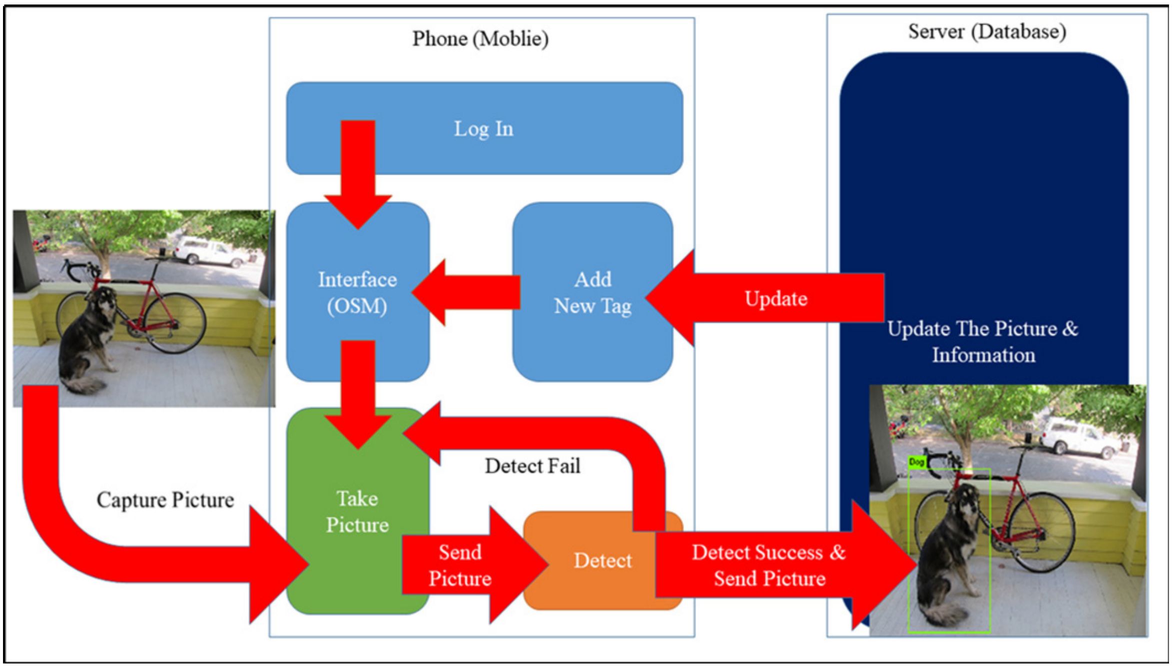

Fig. 4. The system diagram

The reason we don't use a fixed camera is the height. The stray animal is too small in the image, and there are too few features to be acquired. If it is shot from a low place, the angle of view will be disturbed by other objects. The image obtained will be incomplete. And the phone will not have the above problems; you can move it anywhere.

The way we preprocess data is to classify images into cats and dogs. Then the cat and dog appearing on each picture manually mark the rectangle and produce the coordinates of the box on the picture. Finally, Yolo uses pictures and coordinates to train. The Yolo version is divided into Yolo, Yolov2, and Yolov3. Our experiments were mainly trained in Yolov2 and Yolov3. The results of the training will be mentioned in the next section.

4 Results

We used YOLO to do three experiments. The first time we used only 200 pictures of cats and dogs in the dataset, and trained with Yolo v2. The training took seven days. The result was 92% for precision and 22% for recall. The average IOU was 76.53%. In the second experiment, the pictures of cats and dogs were increased to 1,440 sheets, training for two days, Precision reached 80%, Recall was 23%, and Average IOU was 82.57%. However, the experiment used yolov3 for training and 1440 pictures of cats and dogs. Precision can reach 95% in two days, Recall is 22%, and the average IOU is 83.93% (See Table 1).

Table 1. Training Result

	Yolov2	Yolov2	Yolov3
Precision	92%	80%	95%
Recall	22%	23%	22%
Image (Dog)	200	1440	1440
Image (Cat)	200	1440	1440
Training Time	7 days	2 days	2 days
Average IOU	76.53%	82.57%	83.93%

The calculation formulas of Precision and Recall mentioned in Table 1 can be learned from (1) and (2). The confusion matrix in Table 2 is the difference between the actual picture and the predicted picture. If there is a dog in the picture, it means Positive, and if it is not, it means Negative. So, the actual picture has a dog, and the predicted picture is also there, for True Positive. The picture predicts that there is a dog, but it is not, it is False Positive. There are no dogs in the actual picture, and there is no predicted picture. It is True Negative. The real picture has a dog, but it is not anticipated, it is False Negative.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad \textbf{(1)}$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad \textbf{(2)}$$

Table 2. Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The IOU (Intersection-over-union) in the Average IOU can refer to Fig. 5. The IOU is the same as whether the intersection and the union in the actual picture and the predicted picture are close. If the intersection is close to the union, the accuracy of the prediction will be higher. If the gap between the intersection and the union is too large, the accuracy of the prediction will be lower.

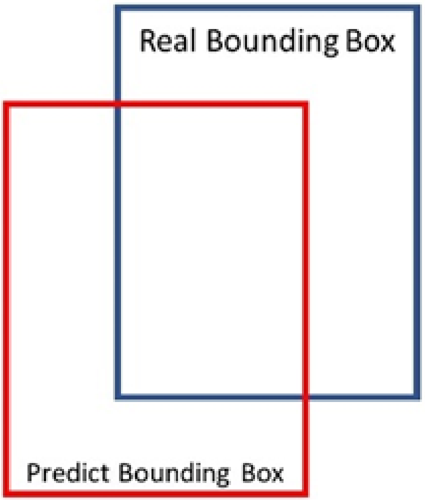

Fig. 5. Intersection-over-union: Real Bounding Box (blue box) between the intersection Predict Bounding Box (red box) the proportion of union

Fig. 6 and Fig. 7 show that the module can successfully detect cats and dogs through the use of yolov3 training. However, only a few pictures of a dog or more than a cat can completely distinguish all the dogs or cats. We also try to test the dog and the cat in the same picture, and only a few (See Fig. 8).

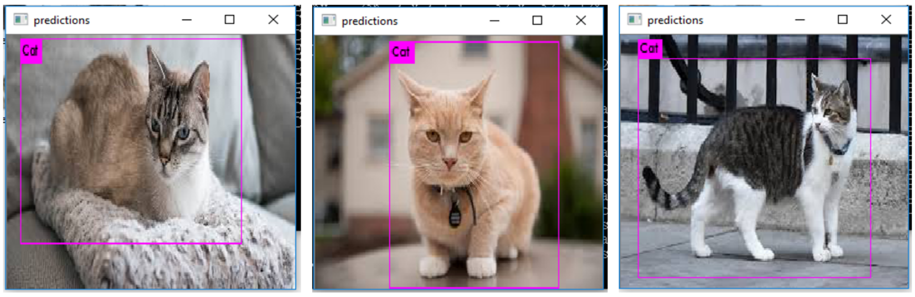

Fig. 6. Cats detected

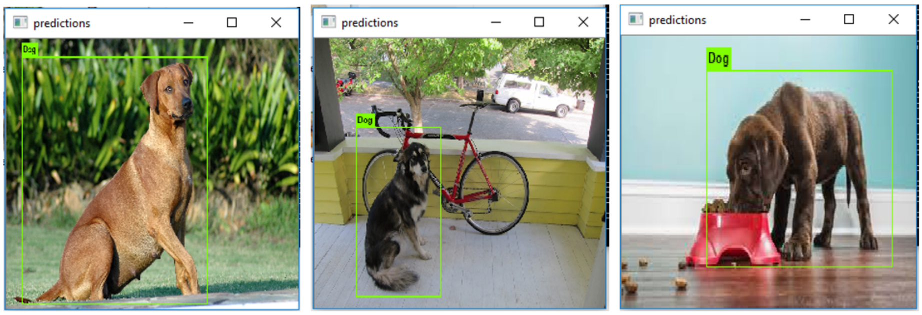

Fig. 7. Dogs detected

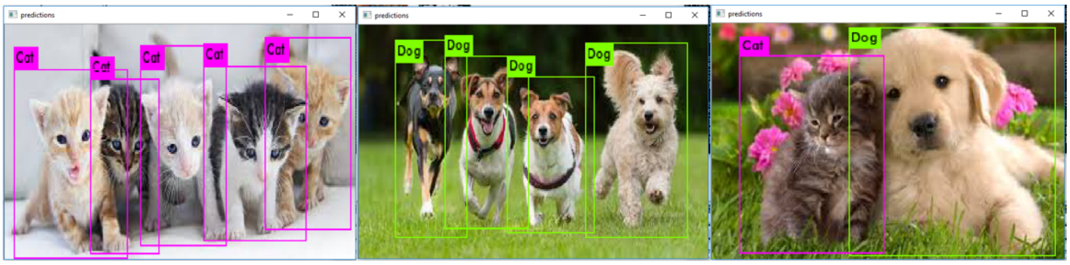

Fig. 8. Multiple object detection (many cats, many dogs, or both in one picture)

5 Conclusion & Future Work

In this system, we use data sets and Yolo v2 to do training first. Next, more data sets and Yolo v3 are used for training. The latter precision is higher. However, there are only a few correct answers for detecting more than two animals in the same picture. Because there is only one animal (cat or dog) for each picture in the training data set that we use, we want users of this system to have expertise in stray animals. When users encounter stray animals, users take photos and upload them to the database. So, all users who use the system can know the location of the stray animals and related assistance information (See Fig. 9). Shelters or pet shops to provide appropriate assistance to reduce the problems extended by stray animals can also use our app.

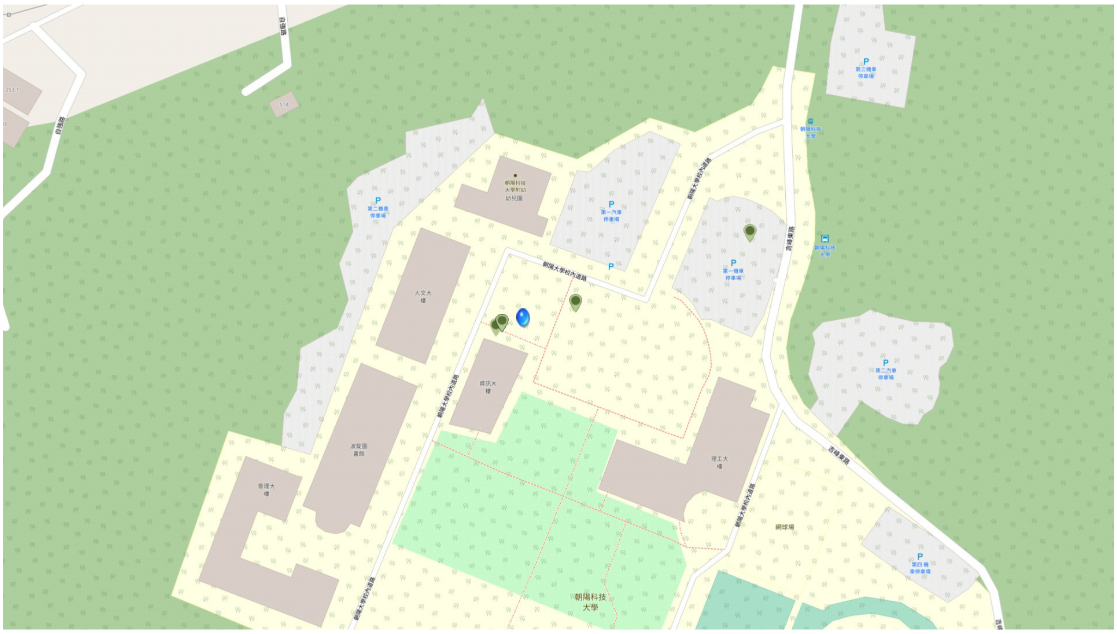

Fig. 9. OSM on our system and green tags are animal locations; the blue spot is the user's location

Besides, we have proposed the following prospects: (1) Use GPS location and information with big data analyze from our database. (2) Use the picture to build a new learning module, which is from a mobile camera or our database. (3) Create game interactions in the app through unity. (4) Tracking wildlife conservation.

Acknowledgments

This paper is supported by the Ministry of Science and Technology, Taiwan. The Nos are MOST-107-2221-E-324 -018 -MY2 and MOST-106-2218-E-324 -002, Taiwan. This research is also partially sponsored by Chaoyang University of Technology (CYUT) and Higher Education Sprout Project, Ministry of Education, Taiwan, under the project name: “The R&D and the cultivation of talent for Health-Enhancement Products.”

References

[1] W.L. Kuo, Y.Y. He, C.K. Chang, Development of a simulation learning environment for inquiry-based learning: an example of stray dogs problem in Taiwan, in: Proc. 2012 IEEE Fourth International Conference on Digital Game And Intelligent Toy Enhanced Learning, Takamatsu, 2012.

[2] C.H. Huang, P.Y. Shen, and Y.C. Huang, Iot-based physiological and environmental monitoring system in animal shelter, in: Proc. 2015 Seventh International Conference on Ubiquitous and Future Networks, Sapporo, 2015.

[3] J. Bright, S.D. Sabbata, S. Lee, B. Ganesh, and D.K. Humphreys, Openstreetmap data for alcohol research: reliability assessment and quality indicators, Health & Place 50(15)(2018) 130-136.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: unified, real-time object detection, in: Proc. the 29th IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016.

[5] Hendry, and R.C. Chen, Automatic license plate recognition via sliding-window darknet-yolo deep learning, Image and Vision Computing 87(4)(2019) 47-56.

[6] P. Pons, J. Jaen, and A. Catala, Assessing machine learning classifiers for the detection of animals’ behavior using depth-based tracking, Expert Systems with Applications 86(22)(2017) 235-246.

[7] J. Monteiro, J.P. Aires, R. Granada, R.C. Barros, and F. Meneguzzi, Virtual guide dog: an application to support visually-impaired people through deep convolutional neural networks, in: Proc. 2017 International Joint Conference on Neural Networks, Anchorage, 2017.

[8] R. Kolandaisamy, K. Subramaniam, I. Kolandaisamy, and S.L. Lin, Stray animal mobile app, in Proc: 2016 Regional Conference On Sciences, Technology And Social Sciences, Pahang, 2016.

[9] J. Xu, G. Solmaz, R. Rahmatizadeh, L. Boloni and D.. Turgut, Providing distribution estimation for animal tracking with unmanned aerial vehicles, in: Proc. 2018 IEEE Global Communications Conference (GLOBECOM), 2018.

[10] X. Zhang, L. Yang, and R. Sinnott, A mobile application for cat detection and breed recognition based on deep learning, in: Proc. 2019 IEEE 1st International Workshop on Artificial Intelligence for Mobile (AI4Mobile), 2019.

[11] C. Deng, Multi user support for senior citizen visual guidance system, [thesis] Department of Information Processing Science, University of Oulu, 2016.

[12] A. Rosebrock, The Keras Blog. blog.keras.io. < https://blog.keras.io/>, 2018 (Accessed 18.01.29).