Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Predicting Canine Posture with Smart Camera Networks Powered by the Artificial Intelligence of Things

Ming-Fong Tsai and Jhao-Yang Huang
Department of Electronic Engineering, National United University, Maioli, Taiwan

Corresponding author: Ming-Fong Tsai (mingfongtsai@gmail.com)

ABSTRACT In today's society, the number of people rearing pets has increased and their awareness of the need to protect pets' health has increased. Pet posture behaviour analysis and prediction are providing assistance in the medical treatment of pets. Hence, the demand for pet skeleton drawing applications has risen dramatically. Our proposed system predicts pet posture using smart camera networks powered by the artificial intelligence of things. This system is built on a platform using a Raspberry Pi embedded system. The system can determine from an image whether there is a detection target and generate a contour mask based on Mask R-CNN Technology. According to object detection, poses and key parts can be identified to predict and draw pet skeletons. Simultaneously, the behavioural action of a pet can be determined according to continuous skeleton data and then the system will actively inform the owner to perform subsequent processing.

INDEX TERMS: Pet Skeletons, Mask R-CNN Technology, Artificial Intelligence of Things;

I. INTRODUCTION

Skeleton drawing technology can be applied to posture and motion recognition or performance capture in posture and motion recognition applications, such as rehabilitation posture, and virtual reality applications, such as making animated films. These applications are usually drawn for human skeletons [1]-[10]. The related work proposes the prediction method of joint points based on deep neural networks [11]. Moreover, the current skeleton drawing method is PoseNet, which uses pictures or videos as input data to detect the skeleton pose of a single person or a group of people [12]. PoseNet uses a pre-trained model to detect a person's 17 key points and features, such as eyes, nose, ears, shoulders, elbows, wrists, hips, knees and ankles. Human skeleton drawing connects lines through the correlation between the above key points. For example, the left hand skeleton will be connected by the left shoulder to the left elbow and then to the left wrist, and the right hand skeleton will be connected by the right shoulder to the right elbow and then to the right wrist. The body skeleton is formed by connecting the left and right shoulders and the left and right hips to form a rectangle. The left foot skeleton is connected by the left hip to the left knee and then to the left ankle, and the right foot skeleton is connected by the right hip to the right knee and then to the right ankle.

In today's society, the number of people rearing pets has increased and their awareness of the need to protect pets' health has increased. Pet posture behaviour analysis and prediction are providing assistance in the medical treatment of pets. Moreover, as the number of pets has exceeded the number of humans, the demand for pet skeleton drawing applications has risen dramatically. The technique used for the skeleton drawing of pets is currently the RGBD method [13], which collects posture information using sensor clothing. However, the use of resources in information collection and skeleton drawing is higher in the RGBD method, as is the calculation complexity. In order to determine the skeleton position and mark the position information of the target object, the related work involves the target object wearing a motion capture suit with a sensor. The Vicon system in Kinect v2 has 20 infrared cameras that are used to record the markers on the dogs' bespoke capture suits. After obtaining the above data, pre-processing of the data is performed and it is input into a 2-stacked hourglass network [14]. In the above network, the optimiser uses RMSprop and the loss function uses the Mean Square Error between the ground truth and network-generated heatmaps to train and obtain the skeleton of the target object. However, to obtain training data from a dog wearing a motion capture suit is not easy and is expensive.

This study proposes a new system that predicts pet posture using smart camera networks powered by the artificial intelligence of things. The system uses a webcam to capture real-time images for pose recognition. In this system, pets do not need to wear sensing elements for sensing clothing. The system is a low-cost and non-invasive computer vision technology, which greatly reduces the time and complexity of data acquisition. By using Mask R-CNN [15] to generate contour mask images as input data for object detection, the impact of image recognition on the background environment is reduced and the accuracy of the object detection is improved. The skeleton drawing of the pet pose prediction system is implemented with eight key points. The skeleton is drawn by connecting lines according to the dependence of these key points. The action state recognition of the pet pose prediction system is based on the use of continuous skeleton information to judge the behaviour of the pet. The main contribution of this paper is to propose a posture predicting method based on Faster R-CNN and Mask R-CNN object detection frameworks. The posture prediction accuracy of the proposed system is 80% in pets and can be easily implemented for posture prediction model training in different species.

The second section of this work gives an introduction to the architecture, process and algorithm of the pet pose prediction system. The third section describes the system experiment platform environment and experimental efficiency results. Finally, the fourth section presents the conclusion and future work.

II. PROPOSED METHOD

A. Overview

The system overview is shown in Figure 1 and the image is captured by a smart network camera. The motion status is recognised based on continuous skeleton information and use real-time images for object detection and skeleton generation. When a specific action state occurs, the owner is notified via communication software.

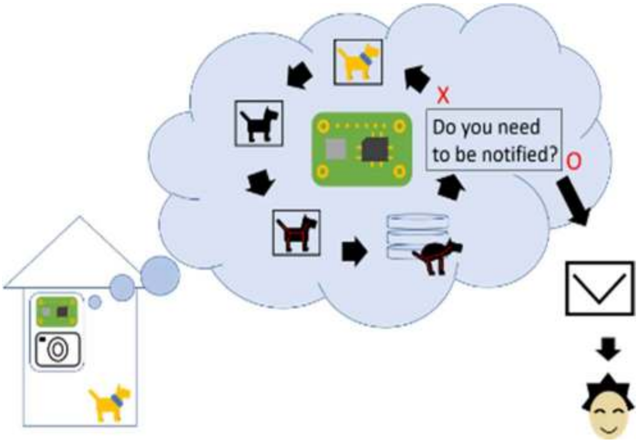


FIGURE 1. **Predicting pet posture system overview**

B. Architecture

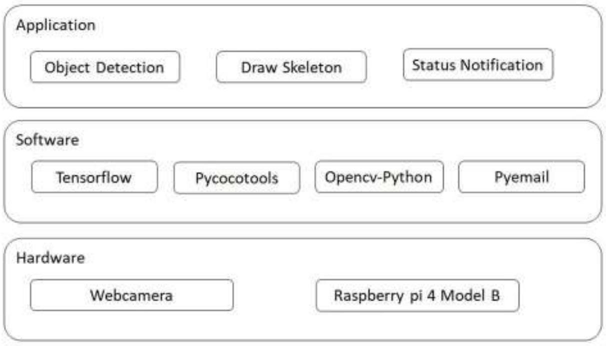


FIGURE 2. **Predicting pet posture system architecture diagram**

The system architecture diagram shown in Figure 2 is divided into hardware, software and application layers. The hardware layer mainly uses a webcam to take photographs and the Raspberry Pi for data calculation and analysis of the core platform. The software layer mainly uses TensorFlow as a machine learning development environment, pycocotools as the Mask R-CNN suite tool, OpenCV-Python for image display and storage, and PyEmail for the email suite tool. The application layer has the functions of specific object detection, skeleton analysis drawing and information notification. The system network architecture diagram is shown in Figure 3. The pre-processing part of the data involves splitting the video of the dog obtained by the webcam into pictures. The split image uses Mask R-CNN to generate the contour mask map of the pet and Faster R-CNN to recognise the posture result of the pet. Moreover, the split image uses Faster R-CNN to identify the key parts of the pet and the position of the corresponding labels. The action state analysis part then uses the position of the pet's key parts and posture to obtain its skeleton key points. The proposed method is combined with the contour mask image output by Mask R-CNN to correct the pet skeleton key points and output a diagram of the pet's skeleton. Finally, the behavioural action of the pet can be determined according to the skeleton diagram.

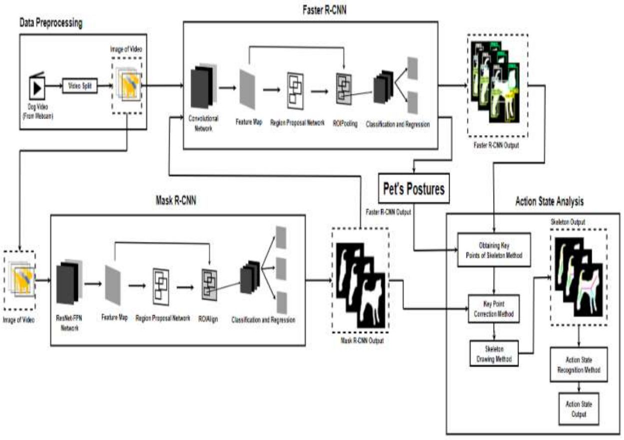


FIGURE 3. **Predicting pet posture system network architecture**

The system flow chart shown in Figure 4 is divided into user, hardware and software sides. The user side has a pet canine and the user's smart handheld device. The pet canine is the target of the judged behaviour. The function of the smart handheld device is to receive action notifications. The hardware side has a smart webcam and an embedded system platform of a Raspberry Pi. The smart webcam is used to capture pet canine images and inputs images to the embedded system platform of the Raspberry Pi for identification and notification. The software side is the environment and kit tool used in the embedded system platform of the Raspberry Pi. Pet canine images identify key parts and poses through the TensorFlow machine learning development environment. The contour mask is generated by the Mask R-CNN function of the pycocotools kit tool. OpenCV-Python is then used to draw the skeleton and save the action state to the database. After the action state is recognised, if it is a specific action state, it will be notified to the owner's smart handheld device via the email package PyEmail.

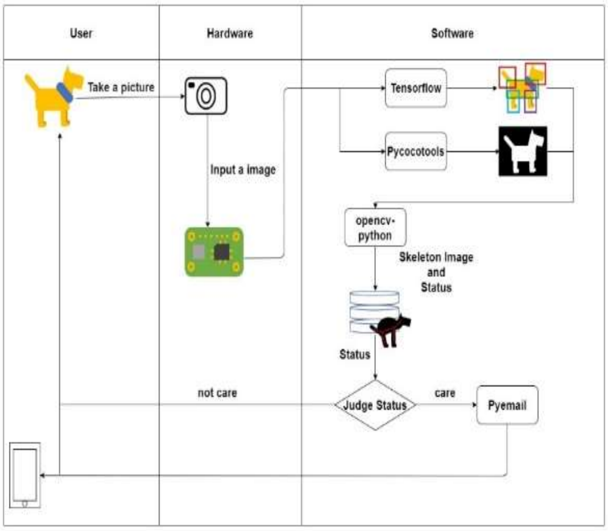
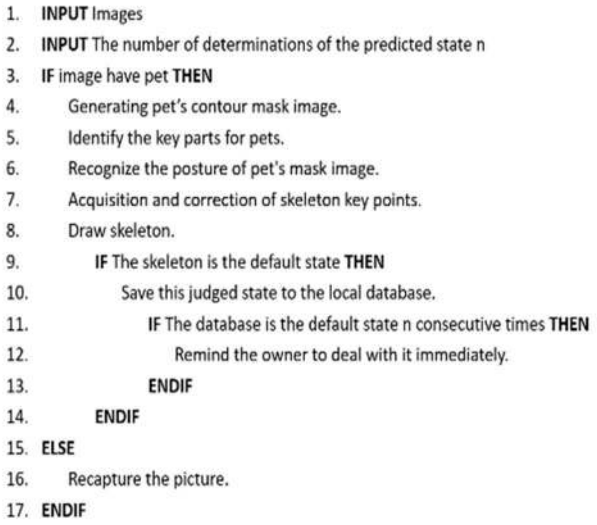


FIGURE 4. **Predicting pet posture system flowchart**

C. System

The main function of the system is shown in Figure 5. The webcam captures the image frames using the Raspberry Pi as a smart network terminal device for image analysis and drawing the skeleton, and the continuous skeleton information is then analysed to identify its action state. When the pet's action posture is in a pre-set state, such as excretion, the owner is notified to clean up. The system performs Mask R-CNN object detection for each captured frame and obtains a contour mask image for each captured frame. A faster R-CNN [16] performs posture recognition on the contour mask image and key part recognition on the original image. After the system successfully obtains the posture analysis results and key part information, the key points of the skeleton are obtained and corrected. A skeleton is drawn based on the dependence of the key points and continuous poses are analysed to judge actions. According to the warning posture set by the system, the owner is reminded to carry out immediate processing.



```
INPUT Images
INPUT The number of determinations of the predicted state n
IF image have pet THEN
    Generating pet's contour mask image.
    Identify the key parts for pets.
    Recognize the posture of pet's mask image.
    Acquisition and correction of skeleton key points.
    Draw skeleton.
        IF The skeleton is the default state THEN
            Save this judged state to the local database.
                IF The database is the default state n consecutive times THEN
                    Remind the owner to deal with it immediately.
                ENDIF
        ENDIF
ELSE
    Recapture the picture.
ENDIF
```

FIGURE 5. **Main function of predicting pet posture system**

1) TO GENERATE THE CONTOUR MASK MAP

Based on the Mask R-CNN object detection, the pet identification and contour mask generation are shown in Figure 6. The object identification weight file is the official version of mask_rcnn_coco.h5. The contour mask images generated by Mask R-CNN are used as input data for posture recognition.

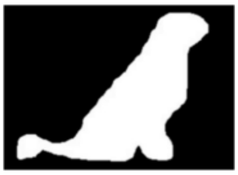


FIGURE 6. **Contour mask image**

2) TO IDENTIFY POSTURE AND KEY PARTS

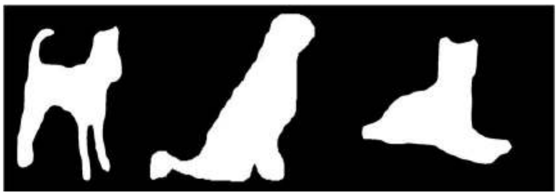


FIGURE 7. **Posture definition**

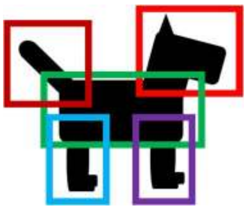
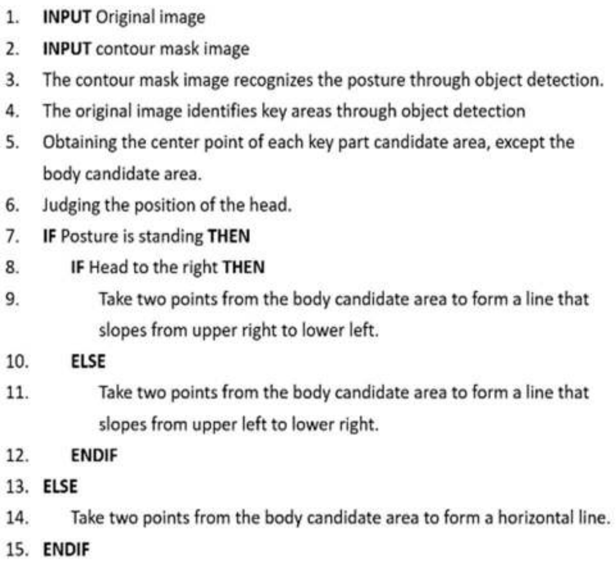


FIGURE 8. **Location of key parts**

The system defines three types of pet poses, namely, standing, sitting and lying, as shown in Figure 7. The system uses the Faster R-CNN network architecture for posture and key part recognition. The posture recognition weight file must be labelled with 320 contour mask images, including the above three pose categories. The posture recognition weight is generated by training a deep learning recognition model. The identification of the key parts is used to define features that help to draw the skeleton, such as the head, front and back feet, body and tail of the pet. The location of the above key parts is obtained through object detection technology, as shown in Figure 8. The recognition weight file of key parts must be labelled with 395 original images, including the above five types of key part and the above three types of posture category original images. The recognition weight file of key parts is generated by training a deep learning recognition model.

D. ALGORITHMS

1) OBTAINING KEY POINTS OF SKELETON



```
INPUT Original image
INPUT contour mask image
The contour mask image recognizes the posture through object detection.
The original image identifies key areas through object detection
Obtaining the center point of each key part candidate area, except the
body candidate area.
Judging the position of the head.
IF Posture is standing THEN
    IF Head to the right THEN
        Take two points from the body candidate area to form a line that
        slopes from upper right to lower left.
    ELSE
        Take two points from the body candidate area to form a line that
        slopes from upper left to lower right.
    ENDIF
ELSE
    Take two points from the body candidate area to form a horizontal line.
ENDIF
```

FIGURE 9. **Key points of skeleton**

The skeleton key point algorithm of this system is shown in Figure 9. The algorithm performs object detection on the original image of the pet to obtain candidate regions of key parts and calculates eight defined key points of the pose. The eight defined key points are the headkeypoint, forebodykeypoint, hindbodykeypoint, forelegkeypoint, foreleg2keypoint, hindlegkeypoint, hindleg2keypoint and tailkeypoint. The candidate area of each key part is an array containing four element values of y_{min}, x_{min}, y_{max}, and x_{max}. Since the pet has two front feet and two rear feet, each of the above key points needs to use object detection to obtain the top two candidate regions of the key part of similarity. The key points use object detection to obtain candidate regions of key parts with the highest similarity, such as the head, front body, back body and tail. The candidate regions of these key parts use the centre point as the key point of the skeleton, such as the head, front legs, back legs and tail. However, the key points of the skeleton of the front body and the back body will be different depending on the pet's posture and head direction. When the X coordinate value of the key point of the head is greater than or equal to half the width of the image, the pet's head will be determined to be on the right. If the X coordinate value of the key point of the head is less than half the width of the image, the pet's head will be determined to be on the left. The contour mask is used for posture recognition based on the Faster R-CNN network architecture. In the element value array of the body candidate area, y_{min} and x_{min} represent the coordinates of the upper left corner of the area box and y_{max} x_{max} represent the coordinates of the lower right corner of the area box.

When the key point of the fore body is at the upper right of the body candidate area frame, y_{min} will move down by a fraction of α of the height of the entire body candidate area frame ($y_{max}-y_{min}$) and x_{max} will move left by a fraction of α of the height of the entire body candidate area frame ($x_{max}-x_{min}$). Similarly, when the key points of the back body are located in the lower left corner of the box of the body candidate area, y_{max} will move up by a fraction of α of the height of the entire body candidate area frame ($y_{max}-y_{min}$) and x_{min} will move right by a fraction of α of the height of the entire body candidate area frame ($x_{max}-x_{min}$). When the pet's head is facing the right and the posture is sitting, the body line will appear to be inclined from the upper right to the lower left. Therefore, the X_f and Y_f coordinates of the fore body key points are obtained using Eqs. (1) and (2), respectively, while the X_h and Y_h coordinates of the back body key points are obtained through Eqs. (3) and (4), respectively. When the pet's head is facing the left and the posture is sitting, the body line will appear to be inclined from the upper left to the lower right. Therefore, the X_f and Y_f coordinates of the fore body key points are obtained using Eqs. (5) and (6), respectively, while the X_h and Y_h coordinates of the back body key points are obtained through Eqs. (7) and (8), respectively.

$$X_f = x_{max} - \frac{x_{max}-x_{min}}{\alpha} \tag{1}$$

$$Y_f = y_{min} + \frac{y_{max}-y_{min}}{a} \tag{2}$$

$$X_h = x_{min} + \frac{x_{max}-x_{min}}{a} \tag{3}$$

$$Y_h = y_{max} - \frac{y_{max}-y_{min}}{a} \tag{4}$$

$$X_f = x_{min} + \frac{x_{max}-x_{min}}{a} \tag{5}$$

$$Y_f = y_{min} + \frac{y_{max}-y_{min}}{a} \tag{6}$$

$$X_h = x_{max} - \frac{x_{max}-x_{min}}{a} \tag{7}$$

$$Y_h = y_{max} - \frac{y_{max}-y_{min}}{a} \tag{8}$$

When the key point of the fore body is at the upper left of the body candidate area frame, y_{min} will move down by a fraction of α of the height of the entire body candidate area frame ($y_{max} - y_{min}$) and x_{min} will move right by a fraction of α of the width of the entire body candidate area frame ($x_{max} - x_{min}$). Similarly, when the key points of the back body are located in the lower right corner of the box of the body candidate area, y_{max} will move up by a fraction of α of the height of the entire body candidate area frame ($y_{max} - y_{min}$) and x_{max} will move left by a fraction of α of the width of the entire body candidate area frame ($x_{max} - x_{min}$). When the pet's head is facing to the right, the body line is horizontal when the posture is lying and standing. The X_f and Y_f coordinates of the fore body key point are obtained by Eqs. (9) and (10), respectively. The X_h and Y_h coordinates of the back body key point are obtained by Eqs. (11) and (12), respectively.

$$X_f = x_{max} - \frac{x_{max} - x_{min}}{a} \tag{9}$$

$$Y_f = y_{min} + \frac{y_{max} - y_{min}}{b} \tag{10}$$

$$X_h = x_{min} + \frac{x_{max} - x_{min}}{a} \tag{11}$$

$$Y_h = y_{min} + \frac{y_{max} - y_{min}}{b} \tag{12}$$

When the key points of the fore body are located in the box of the body candidate area, y_{min} will move down by a fraction of b of the height of the entire body candidate area frame ($y_{max} - y_{min}$) and x_{max} will move left by a fraction of α of the width of the entire body candidate area frame ($x_{max} - x_{min}$). When the key points of the fore body are located in the box of the body candidate area and the lines of the body appear horizontal, y_{min} will move down by a fraction of b of the height of the entire body candidate area frame ($y_{max} - y_{min}$) and x_{min} will move right by a fraction of α of the width of the entire body candidate area frame ($x_{max} - x_{min}$). When the pet's head is facing to the left, the body line is horizontal when the posture is lying and standing. The X_f and Y_f coordinates of the fore body key point are obtained by Eqs. (13) and (14), respectively. The X_h and Y_h coordinates of the back body key point are obtained by Eqs. (15) and (16), respectively. This study designs "a and b ∈ C" and sets α to a constant value of 4 and b to a constant value of 2.

$$X_f = x_{min} + \frac{x_{max} - x_{min}}{a} \tag{13}$$

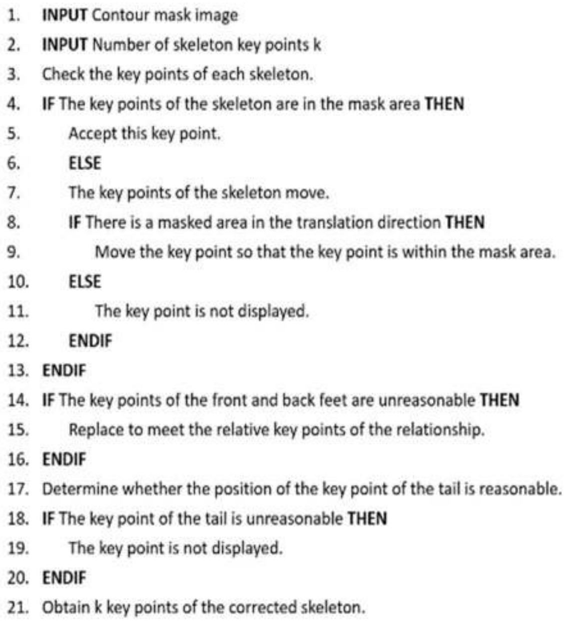
$$Y_f = y_{min} + \frac{y_{max} - y_{min}}{b} \tag{14}$$

$$X_h = x_{max} - \frac{x_{max} - x_{min}}{a} \tag{15}$$

$$Y_h = y_{min} + \frac{y_{max} - y_{min}}{b} \tag{16}$$

2) ALGORITHM FOR KEY POINT CORRECTION

The key point correction algorithm is shown in Figure 10 to ensure that the key points of the skeleton are in the contour mask. When the key points of the front and back foots are not in the contour mask, the horizontal translation is used for correction. When the head is facing to the left, the key point will move to the right into the contour mask. Conversely, we pan to the left into the contour mask to complete the key point correction. When the tail key point is not in the contour mask, we use vertical movement to correct. When there is no contour mask on the upper and lower sides of the original key point, the key point will not be displayed. The position of the key points of the revised front and back foots needs to meet the relative relationship between the front and back foots. If any one of the key points of the front or back foot does not meet the above conditions, the key points that meet the relative relationship between the front and back will be replaced.



```
INPUT Contour mask image
INPUT Number of skeleton key points k
Check the key points of each skeleton.
IF The key points of the skeleton are in the mask area THEN
    Accept this key point.
    ELSE
    The key points of the skeleton move.
    IF There is a masked area in the translation direction THEN
        Move the key point so that the key point is within the mask area.
    ELSE
        The key point is not displayed.
    ENDIF
ENDIF
IF The key points of the front and back feet are unreasonable THEN
    Replace to meet the relative key points of the relationship.
ENDIF
Determine whether the position of the key point of the tail is reasonable.
IF The key point of the tail is unreasonable THEN
    The key point is not displayed.
ENDIF
Obtain k key points of the corrected skeleton.
```

FIGURE 10. **Key point correction**

After the correction, the key position of the tail must meet the following calculation. When the head is facing to the left, the value of the X coordinate of the tail key point after the correction needs to meet the calculation result of Eq. (17). Compliance with the calculation result indicates that the key point of the tail is at a reasonable position; otherwise, the key point will not be displayed. When the head is facing to the right, the value of the X coordinate of the tail key point after the correction needs to meet the calculation result of Eq. (18). Compliance with the calculation result indicates that the key point of the tail is at a reasonable position; otherwise, the key point will not be displayed. When α is set to a value of 1, the tail key points are easy to draw but the error rate is high. As the value of α increases, the drawing rate will be lower but the accuracy rate will increase. This study sets α to a value of 3. After obtaining the key point correction of the tail according to the object detection, we determine whether its position is in the contour mask. If the above method is not established, vertical translation correction on the key points must be performed. When there is no contour mask above and below the original key, the key will not be displayed. Finally, according to Eqs. (17) and (18), we determine whether the key point of the tail after translation is at a reasonable position.

$$X_{tailkeypoint} \geq X_{hindbodykeypoi} - \frac{X_{hindbodykeypoint} - X_{forebodykeypoint}}{\alpha}, \alpha \geq 1 \quad (17)$$

$$X_{tailkeypoint} \leq X_{hindbodykeypoin} + \frac{X_{forebodykeypoint} - X_{hindbodykeyp}}{\alpha}, \alpha \geq 1 \quad (18)$$

3) ALGORITHM FOR SKELETON DRAWING

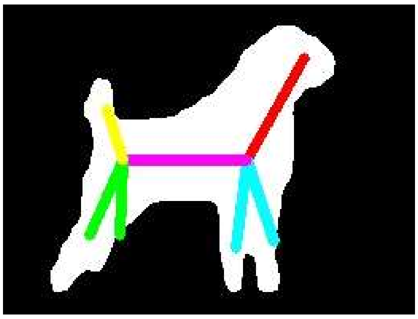
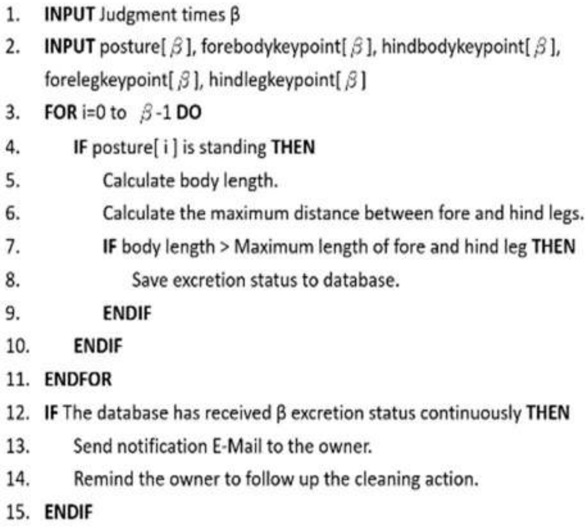


FIGURE 11. **Skeleton drawing**

We connect the eight skeleton key points of the pet according to the key point dependence, as shown in Figure 11. The key point dependencies are the head corresponding to the front body, the front body corresponding to the front foot, the front body corresponding to the back body, the back body corresponding to the back foot and the back body corresponding to the tail key point.

4) ALGORITHM OF ACTION STATE RECOGNITION



```
INPUT Judgment times β
INPUT posture[ β ], forebodykeypoint[ β ], hindbodykeypoint[ β ],
forelegkeypoint[ β ], hindlegkeypoint[ β ]
FOR i=0 to  β-1 DO
    IF posture[ i ] is standing THEN
        Calculate body length.
        Calculate the maximum distance between fore and hind legs.
        IF body length > Maximum length of fore and hind leg THEN
            Save excretion status to database.
        ENDIF
    ENDIF
ENDFOR
IF The database has received β excretion status continuously THEN
    Send notification E-Mail to the owner.
    Remind the owner to follow up the cleaning action.
ENDIF
```

FIGURE 12. **Action state recognition**

The action state recognition algorithm of this system takes pet excretion as an example, as shown in Figure 12. The action contour mask for pet excretion is shown in Figure 13. In posture analysis, it will be determined whether the pet is in the standing category and the key points of the skeleton will be obtained from the front body, back body, front foot, back foot and so on. This information is then used as input information for the motion state recognition algorithm. The distance between the key points of the front and rear foots has a relatively close characteristic based on the state of the excretion action of the pet. If the calculation result of Eq. (19) is met, the pet complies with the excretion action of the pet. When multiple consecutive pictures captured by the intelligent network camera meet the above conditions, it is determined that the pet is in a state of excretion. The system will send a notification to the owner's e-mail to remind them of the follow-up cleaning action.

$$|X_{forebodykeypoint} - X_{hindbodykeypoint}| > |X_{forelegkeypoint} - X_{hindlegkeypoi}| \quad (19)$$

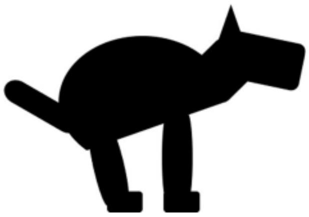


FIGURE 13. **Contour mask of pet excretion action**

III. EVALUATION AND RESULTS

A. EXPERIMENTAL PLATFORM AND ENVIRONMENT

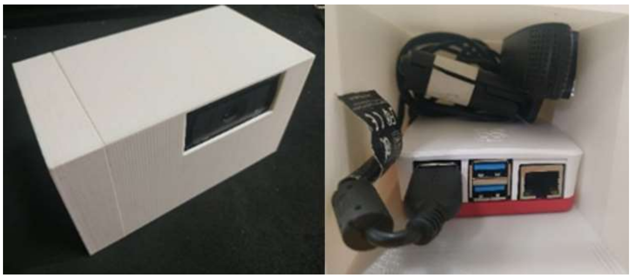


FIGURE 14. **Webcam of the smart Internet of Things**

TABLE I
EXPERIMENTAL PLATFORM

Camera	Logitech WEBCAM C920R
Platform	Raspberry Pi 4 Model B
Programming Language	Python3.7
Main Library	Tensorflow == 1.14.0 Pycocotools == 2.0 OpenCV-Python == 3.4.6.27 PyEmail

The experimental platform information is shown in Table 1. The webcam is a Logitech Webcam C920R. The edge device of the smart Internet of Things is the embedded system Raspberry Pi 4 Model B. The system is written using the Python programming language and a library for the deep learning development environment TensorFlow, the Mask R-CNN library of pycocotools, OpenCV-Python image processing library and the PyEmail library. The system appearance and hardware configuration are shown in Figure 14.

B. ACCURACY ANALYSIS FOR IDENTIFICATION

TABLE II
EXPERIMENTAL RESULTS OF ACCURACY

	TOP 1	TOP 2	TOP 3	TOP 4	TOP 5
Faster R-CNN	0%	0%	66.6%	100%	100%
SSD-MobileNet V1	0%	8%	8%	50%	91%
SSD-MobileNet V2	8%	91.6%	100%	100%	100%
YOLOv3	16.6%	100%	100%	100%	100%
YOLOv4	0%	0%	50%	100%	100%
Proposed Method	83.3%	91.6%	100%	100%	100%

TABLE III
EXPERIMENTAL RESULTS OF ACCURACY WITH CLEAN BACKGROUND

	TOP 1	TOP 2	TOP 3	TOP 4	TOP 5
Faster R-CNN	0%	0%	70%	100%	100%
SSD-MobileNet V1	0%	0%	0%	100%	100%
SSD-MobileNet V2	0%	60%	100%	100%	100%
YOLOv3	30%	100%	100%	100%	100%
YOLOv4	0%	50%	100%	100%	100%
Proposed Method	83.3%	91.6%	100%	100%	100%

The system uses deep learning image recognition methods for pet excretion model training and object detection. The accuracy of Faster R-CNN object detection is shown in Table 2. A recognition model is trained using the Faster R-CNN recognition model, using 525 sample pictures, including four pet poses of standing, sitting, lying and excretion. The identification model is also generated by using 525 sample images, including the four poses of standing, sitting, lying and excretion, and labelling with the above types and using the Faster R-CNN identification model for model training actions. Performance tests are carried out using 12 non-training sample sets of pictures of pet excretion. The accuracy of the Faster R-CNN recognition result puts the excretion status in the first rank as 0% and the accuracy until the recognition result is placed in the third rank rises to 66.67%. The accuracy of SSD_MobileNet V1 [17] object detection is shown in Table 2. A recognition model is trained using the SSD_MobileNet V1 recognition model, using 525 sample pictures, including four pet poses of standing, sitting, lying and excretion. The identification model is also generated by using 525 sample images, including the four poses of standing, sitting, lying and excretion, and labelling with the above types and using the SSD_MobileNet V1 identification model for model training actions. Performance tests are carried out using 12 non-training sample sets of pictures of pet excretion. The accuracy of the SSD_MobileNet V1 recognition result puts the excretion status in the first rank as 0% and the accuracy until the recognition result is placed in the fourth rank rises to 50%. The accuracy of SSD_MobileNet V2 [18] object detection is shown in Table 2. A recognition model is trained using the SSD_MobileNet V2 recognition model, using 525 sample pictures, including four pet poses of standing, sitting, lying and excretion. The identification model is also generated by using 525 sample images, including the four poses of standing, sitting, lying and excretion, and labelling with the above types and using the SSD_MobileNet V2 identification model for model training actions. Performance tests are carried out using 12 non-training sample sets of pictures of pet excretion. The accuracy of the SSD_MobileNet V2 recognition result puts the excretion status in the first rank as 8% and the accuracy until the recognition result is placed in the second rank rises to 91.6%. The accuracy of YOLOv3 [19] object detection is shown in Table 2. A recognition model is trained using the YOLOv3 recognition model, using 525 sample pictures, including four pet poses of standing, sitting, lying and excretion. The identification model is also generated by using 525 sample images, including the four poses of standing, sitting, lying and excretion, and labelling with the above types and using the YOLOv3 identification model for model training actions. Performance tests are carried out using 12 non-training sample sets of pictures of pet excretion. The accuracy of the YOLOv3 recognition result puts the excretion status in the first rank as 16.6% and the accuracy until the recognition result is placed in the second rank rises to 100%. The accuracy of YOLOv4 [20] object detection is shown in Table 2. A recognition model is trained using the YOLOv4 recognition model, using 525 sample pictures, including four pet poses of standing, sitting, lying and excretion. The identification model is also generated by using 525 sample images, including the four poses of standing, sitting, lying and excretion, and labelling with the above types and using the YOLOv4 identification model for model training actions. Performance tests are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3042539, IEEE Access

carried out using 12 non-training sample sets of pictures of pet excretion. The accuracy of the YOLOv4 recognition result puts the excretion status in the third rank as 50% and the accuracy until the recognition result is placed in the fourth rank rises to 100%. The reason for this is that the pet's stance and excretion state are very similar, resulting in poor object detection. The method proposed in this study uses skeleton key point information to distinguish pet similarity states. Therefore, the accuracy of the proposed method's recognition result puts the first state of excretion at 83.33% of the value, until the accuracy of the recognition result is placed in the third order before the accuracy rises to 100%. Therefore, the accuracy of the proposed method's recognition of placing the excretion in the first order is 83.33%. Until the recognition result is placed in the top three, the accuracy increases to 100%.

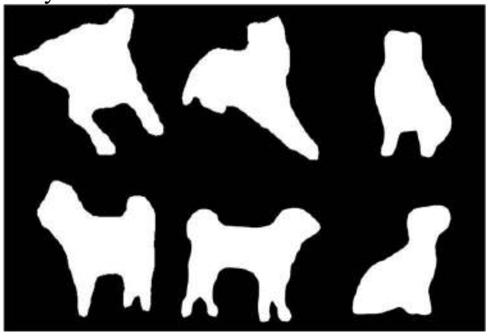


FIGURE 15. Samples of pet poses

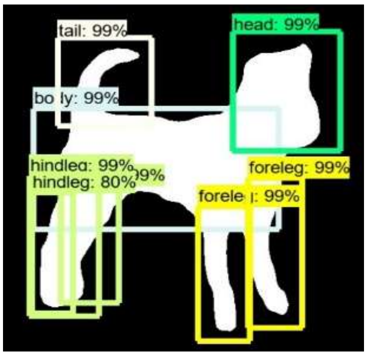


FIGURE 16. Identification results

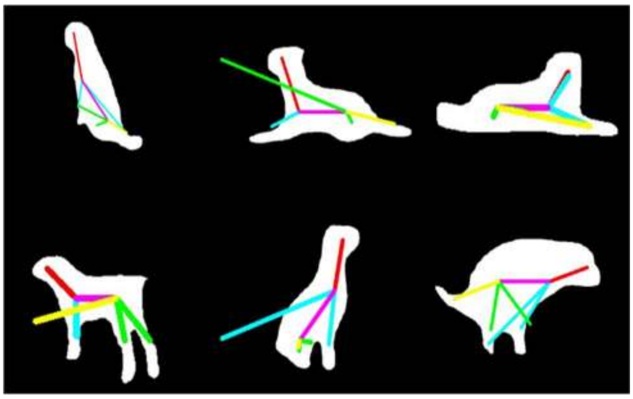


FIGURE 17. Skeleton drawing results without correction

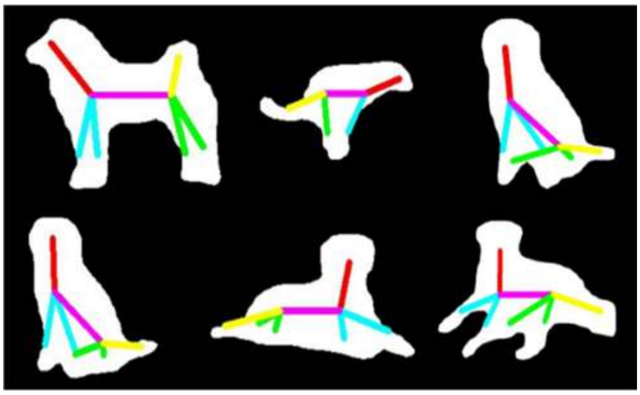


FIGURE 18. Dog's skeleton drawing results

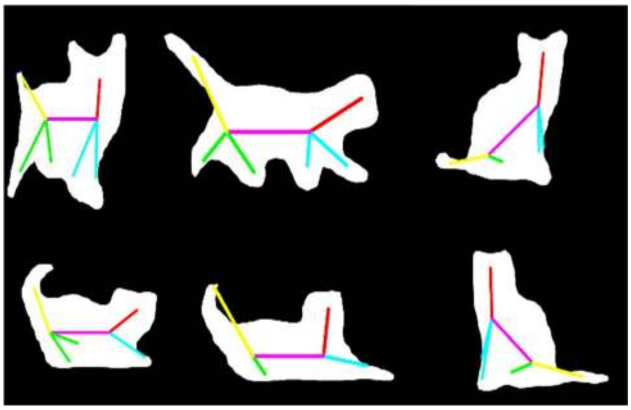
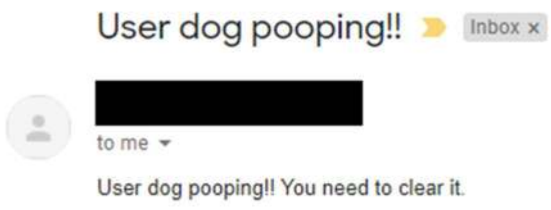


FIGURE 19. Cat's skeleton drawing results



User dog pooping!! Inbox ×

[redacted]

to me

User dog pooping!! You need to clear it.

FIGURE 20. Notification

The system uses deep learning image recognition methods for pet excretion model training and object detection with clean background. The accuracy of Faster R-CNN, SSD_MobileNet V1, SSD_MobileNet V2, YOLOv3, YOLOv4 and proposed method object detection are shown in Table 3. Performance tests are carried out using 10 non-training sample sets of white background pictures of pet excretion. The accuracy of the Faster R-CNN recognition result puts the excretion status in the first rank as 0% and the accuracy until the recognition result is placed in the third rank rises to 70%. The accuracy of the SSD_MobileNet V1 recognition result puts the excretion status in the first rank as 0% and the accuracy until the recognition result is placed in the fourth rank rises to 100%. The accuracy of the SSD_MobileNet V2 recognition result puts the excretion status in the first rank as 0% and the accuracy until the recognition result is placed in the second rank rises to 60%. The accuracy of the YOLOv3 recognition result puts the excretion status in the first rank as 30% and the accuracy until the recognition result is placed in the second rank rises to 100%. The accuracy of the YOLOv4 recognition result puts the excretion status in the second rank as 50% and the accuracy until the recognition result is placed in the third rank rises to 100%. The reason for this is that the pet's stance and excretion state are very similar, resulting in poor object detection. The method proposed in this study uses skeleton key point information to distinguish pet similarity states. Therefore, the accuracy of the proposed method's recognition result puts the first state of excretion at 80% of the value, until the accuracy of the recognition result is placed in the second order before the accuracy rises to 100%. Therefore, the accuracy of the proposed method's recognition of placing the excretion in the first order is 80%. Until the recognition result is placed in the top two, the accuracy increases to 100%.

A pet posture recognition model is generated using network architecture Faster R-CNN training generation. The training samples of the training data set is a contour mask, as shown in Figure 15. The 320 training sample images have three contour mask images: lying posture, sitting posture and standing posture. Using 15 mask images of non-training sample dataset for performance testing, the accuracy is 80%. Pet key part identification model training is generated using the Faster R-CNN network architecture. The label categories of the training data set include five types of labels, namely, head, body, forefoot, hindfoot and tail. The training data set uses 395 pictures to generate a pet key part recognition model. The recognition results are shown in Figure 16. This system uses a smart webcam to capture real-time images. The image uses Mask R-CNN object detection for pet identification and contour mask generation. After using Faster R-CNN to identify poses and key parts, skeleton key points will be obtained. The skeleton key point correction algorithm is not used as a supplement and the skeleton is drawn, as shown in Figure 17. The skeleton key point correction algorithm is used as a supplement and the skeleton is drawn, as shown in Figure 18. The cat's image uses Mask R-CNN object detection for pet identification and contour mask generation. After using Faster R-CNN to identify poses and key parts, skeleton key points will be obtained. The skeleton key point correction algorithm is used as a supplement and the skeleton is drawn, as shown in Figure 19. Figure 18 and Figure 19 contains the results of the pet's standing, sitting, lying and excretion status. In the above results, the red line indicates the head, the blue line indicates the forefoot, the magenta line indicates the body and the green line indicates the rear foot and the yellow line indicates the tail. When the system judges that the pet is in a state of excretion by the action state recognition algorithm, it will immediately send an email to notify the owner for subsequent processing. The notification e-mail is shown in Figure 20. Analysis of the execution time of the proposed system is shown in Figure 21. Experimental tests of the proposed system are performed ten times to capture pictures containing the dog, and the total time of each system execution is recorded. The average execution time is 201.72 seconds in the first test environment, which uses a Raspberry Pi 4 embedded system with 4GB RAM. The average execution time is 69.24 seconds in the second test environment using a Raspberry Pi 4 embedded system with 4GB RAM and accelerator. The accelerator uses the Google Coral USB Accelerator for generation of the contour mask image.

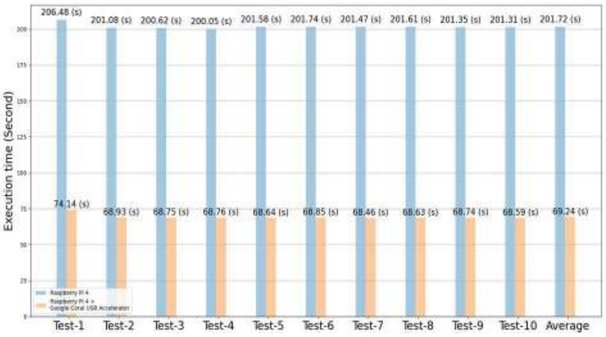


FIGURE 21. Analysis of execution time

IV. CONCLUSIONS

This study has proposed the use of Faster R-CNN for the object detection of pets' key parts. It then uses Mask R-CNN for the object detection of the pet and contour mask generation, and draws the skeleton information and motion status recognition of the pet. This method solves the problem that skeleton drawing is currently aimed at humans and proposes to quickly establish a skeleton drawing process and method for new objects. There is need not to put on clothes with sensors when collecting key points of the skeleton. Only deep learning through the picture is required and by carrying on the training of the image recognition model can achieve the skeleton drawing effect and greatly reduce the operation complexity. This method solves the problem that the accuracy of motion recognition accuracy of object detection is not ideal in a single image. The accuracy of the action recognition is 83.33% in the first order and 100% in accuracy before the third.

REFERENCES

[1] W. Nie, W. Wang and X. Huang, SRNet: Structured Relevance Feature Learning Network from Skeleton Data for Human Action Recognition, IEEE Access, vol. 7, pp. 132161-132172, 2019.

[2] M. Andriluka, L. Pishchulin, P. Gehler and B. Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. IEEE Conference on Computer Vision and Pattern Recognition, pp. 3686-3693, 2014.

[3] M. Dantone, J. Gall, C. Leistner and L. Gool, Human Pose Estimation using Body Parts Dependent Joint

Regressors. IEEE Conference on Computer Vision and Pattern Recognition, pp. 3041-3048, 2013.

[4] T. Zou and T. Sugihara, Fast Identification of a Human Skeleton-marker Model for Motion Capture System using Stochastic Gradient Descent Method, IEEE International Conference for Biomedical Robotics and Biomechatronics, pp. 181-186, 2020.

[5] S. Boulahia, E. Anquetil, R. Kulpa and F. Multon, 3D Multistroke Mapping (3DMM): Transfer of Hand-Drawn Pattern Representation for Skeleton-Based Gesture Recognition, IEEE International Conference on Automatic Face and Gesture Recognition, pp. 462-467, 2017.

[6] F. Juang, T. Chen and W. Du, Human Body 3D Posture Estimation using Significant Points and Two Cameras, The Scientific World Journal, vol. 2014, pp. 1-17, 2014.

[7] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio and A. Blake, Efficient Human Pose Estimation from Single Depth Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 12, pp. 2821-2840, 2012.

[8] M. Cui, J. Fang and Y. Zhao, Emotion Recognition of Human Body's Posture in Open Environment, IEEE Chinese Control and Decision Conference, pp. 3294-3299, 2020.

[9] S. Obdrzalek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison and M. Pavel, Accuracy and Robustness of Kinect Pose Estimation in the Context of Coaching of Elderly Population, International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1188-1193, 2012.

[10] Z. Cao, G. Hidalgo, T. Simon, S. Wei and Y. Sheikh, OpenPose: Realtime Multi-person 2D Pose Estimation using Part Affinity Fields, IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI0.1109/TPAMI.2019.2929257, 2019.

[11] A. Toshev and C. Szegedy. Deeppose: Human Pose Estimation via Deep Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2014.

[12] A. Kendall, M. Grimes and R. Cipolla, PoseNet: A Convolutional Network for Real-time 6-DOF Camera Relocalization, IEEE International Conference on Computer Vision, pp. 2938–2946, 2015.

[13] S. Kearney, W. Li, M. Parsons, K. Kim and D. Cosker, RGBD-Dog: Predicting Canine Pose from RGBD Sensors, EEE Conference on Computer Vision and Pattern Recognition, pp. 8336-8345, 2020.

[14] A. Newell, K. Yang and J. Deng, Stacked Hourglass Networks for Human Pose Estimation, European Conference on Computer Vision, pp. 483-499, 2016.

[15] K. He, G. Gkioxari, P. Dollar and R. Girshick, Mask R-CNN, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-12, 2017.

[16] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 9, pp. 1-9, 2017.

[17] M. Rahman, P. Kapoor, R. Laganiere, D. Laroche, C. Zhu, X. Xu and A. Ors, Deep People Detection: A Comparative Study of SSD and LSTM-decoder, pp. 305-312, 2018.

[18] Y. Chiu, C. Tsai, M. Ruan, G. Shen and T. Lee, Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems, IEEE International Conference on System Science and Engineering, pp. 1-5, 2020.

[19] J. Redmon and A. Farhadi, Yolov3: An Incremental Improvement, IEEE International Conference on Computer Vision, pp. 1–6, 2018.

[20] Y. Li, H. Wang, L. Dang, T. Nguyen, D. Han, A. Lee, I. Jang and H. Moon, A Deep Learning-based Hybrid Framework for Object Detection and Recognition in Autonomous Driving, IEEE Access, DOI: 10.1109/ACCESS.2020.3033289, 2020.

Ming-Fong Tsai received the Ph.D. degree from the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. He is currently an Assistant Professor with the Department of Electronic Engineering, National United University, Taiwan. His current research interests include Internet of Things, Mechanism and Deep Learning Technology, Vehicular Communications and Multimedia Communications.

Jhao-Yang Huang received the B.S. degree from the Department of Electronic Engineering, National United University, Taiwan. His current research interests include Internet of Things and Deep Learning Technology.