Sentiment Analysis of Pets Using Deep Learning Technologies in Artificial Intelligence of Things System

Ming-Fong Tsai (mingfongtsai@gmail.com)
National United University https://orcid.org/0000-0001-9046-2513
Jhao-Yang Huang
National United University

Research Article

Keywords: Sentiment Analysis, Deep Learning, Artificial Intelligence of Things

Posted Date: March 22nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-330317/v1

Version of Record: A version of this preprint was published at Soft Computing on August 5th, 2021. See the published version at https://doi.org/10.1007/s00500-021-06038-z.

Sentiment Analysis of Pets using Deep Learning Technologies in Artificial Intelligence of Things System

Ming-Fong Tsai* and Jhao-Yang Huang
Department of Electronic Engineering, National United University, Miaoli, Taiwan
Corresponding: mftsai@nuu.edu.tw

Abstract
This research paper proposes sentiment analysis of pets using deep learning technologies in the artificial intelligence of things system. Mask R-CNN is used to detect image objects and generate contour mask maps, pose analysis algorithms are used to obtain object posture information, and at the same time object sound signals are converted into spectrograms as features, using deep learning image recognition technology to obtain object emotion information. By using the fusion of object posture and emotional characteristics as the basis for pet emotion identification and analysis, the detected specific pet behaviour states will be actively notified to the owner for processing. Compared with traditional speech recognition, which uses mel-frequency cepstral coefficients for feature extraction, coupled with a Gaussian mixture model-hidden Markov model for voice recognition, the experimental method of this research paper effectively improves the accuracy by 80%.
Keywords: Sentiment Analysis; Deep Learning; Artificial Intelligence of Things;

1. Introduction

Pet sentiment analysis can be used to analyse whether a pet is suffering from anxiety, hypothetical disorders and other mental illnesses. Pet sentiment analysis can also be used to obtain more subtle information where pets hide the emotion. For example, when pets are in a vigilant mood, the hidden subtle messages are the response to strangers or strange objects [1]-[5]. As the number of pets increases, the demand for pet sentiment analysis will increase. Traditional sentiment analysis uses voice recognition to analyse emotions [6]-[9], and this uses mel-frequency cepstral coefficients to extract features of the input audio [10]-[12]. The main process of using MFCC for feature extraction is obtained by performing the following eight steps on the input source. The first step is pre-emphasis, used to highlight the high-frequency formant, and the second step is frame blocking, which combines x sampling points into a sound frame, where x is usually 256 or 512, and the third step is the Hamming window,

which multiplies each sound frame by the Hamming window to increase the continuity between the left and right ends of the sound frame; the fourth step is Fast Fourier Transform [13], the fifth step is triangular bandpass filters, and the sixth step is discrete cosine transform [14]; the seventh step is log energy and the eighth step is the delta cepstrum. After obtaining the features through the above eight steps, a Gaussian mixture model–Hidden Markov model is finally used for speech recognition analysis. Because the MFCC is based on the human ear which can accurately distinguish human speech, it simulates the operation of the human ear in an artificial way, and the main sensitive frequency range of the human ear is 200 Hz to 500 0Hz, so mel-cepstral coefficients are not suitable for processing sounds other than from human. This research paper proposes sentiment analysis of pets using deep learning technologies in the artificial intelligence of things system, using Mask R-CNN to detect and recognise object tags and generate corresponding contour masks to obtain posture features, and uses object sound signals to convert into spectrograms for recognition and analysis to obtain emotions features in order to realise pet emotion analysis through a non-contact smart IoT system. The second chapter explains the pet sentiment analysis system architecture, system process and algorithm of the smart IoT system. The third chapter explains the experimental environment setting and performance analysis of pet sentiment analysis of the smart Internet of Things system. The fourth chapter is the conclusion and recommends future work.

2. Sentiment Analysis of Pets in Artificial Intelligence of Things System

2.1 System Overview

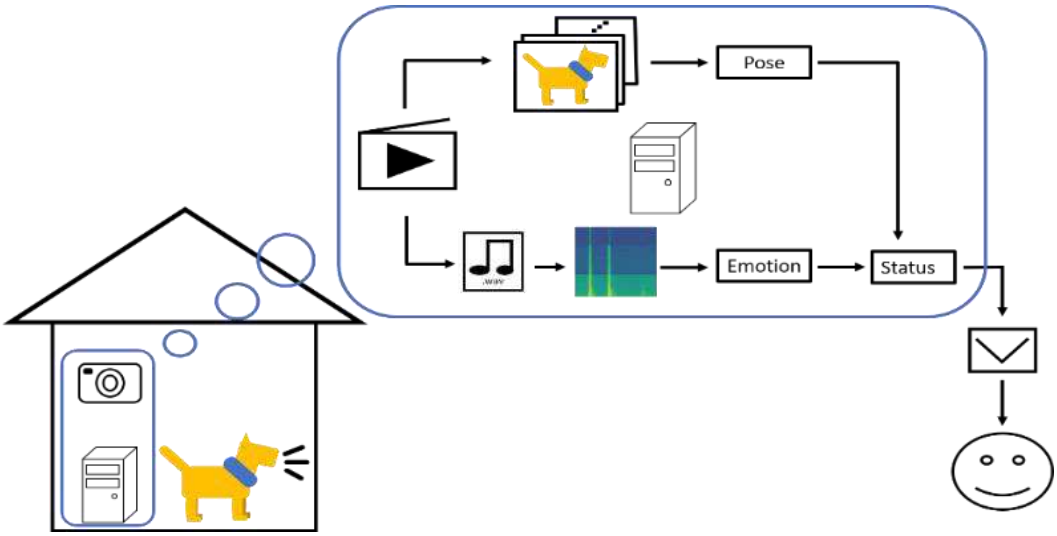

Figure 1: System Overview

The overview of the pet sentiment analysis system of the Smart Internet of Things system is shown in Figure 1. A smart web camera is used to capture pet video and audio information, pet posture analysis for continuous images, and pet sentiment analysis for sound. Pet emotion analysis and recognition is performed based on the posture and emotion information of the above-mentioned deep learning image recognition. When the specific emotional state of the pet is determined, it will be notified to the owner in real time through the communication software for processing.

2.2 System Architecture

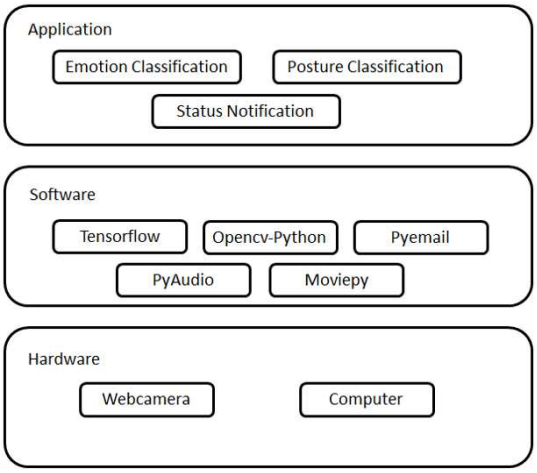

Figure 2: System Architecture Diagram

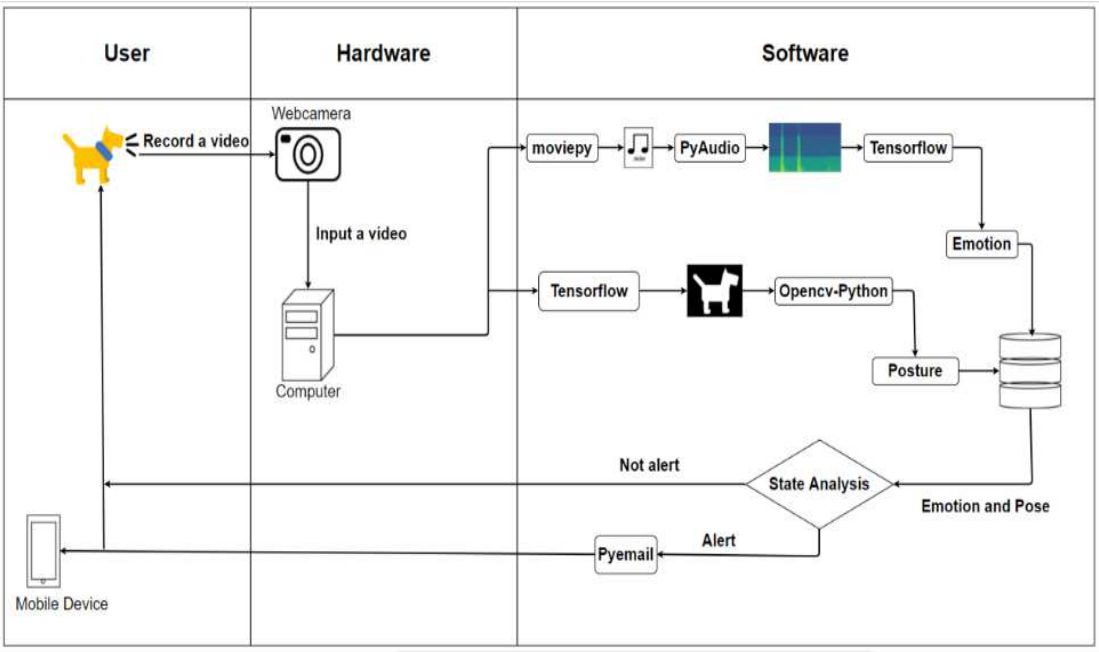

Figure 3: System Fowchart

The structure of the pet sentiment analysis system of the smart IoT system is shown in Figure 2 and consists of three parts: the hardware layer, the software layer and the application layer. The hardware layer mainly uses network cameras for video and audio capture and computing core platforms for data analysis and calculation. The software layer mainly uses the Tensorflow framework as a machine learning development environment platform, OpenCV-Python for image display and storage functions, PyEmail for email processing kit tools, PyAudio for sound file processing kit tools, and MoviePy for sound file extraction and storage. The application layer has functions of deep learning emotion recognition, deep learning gesture recognition and pet specific state notification.

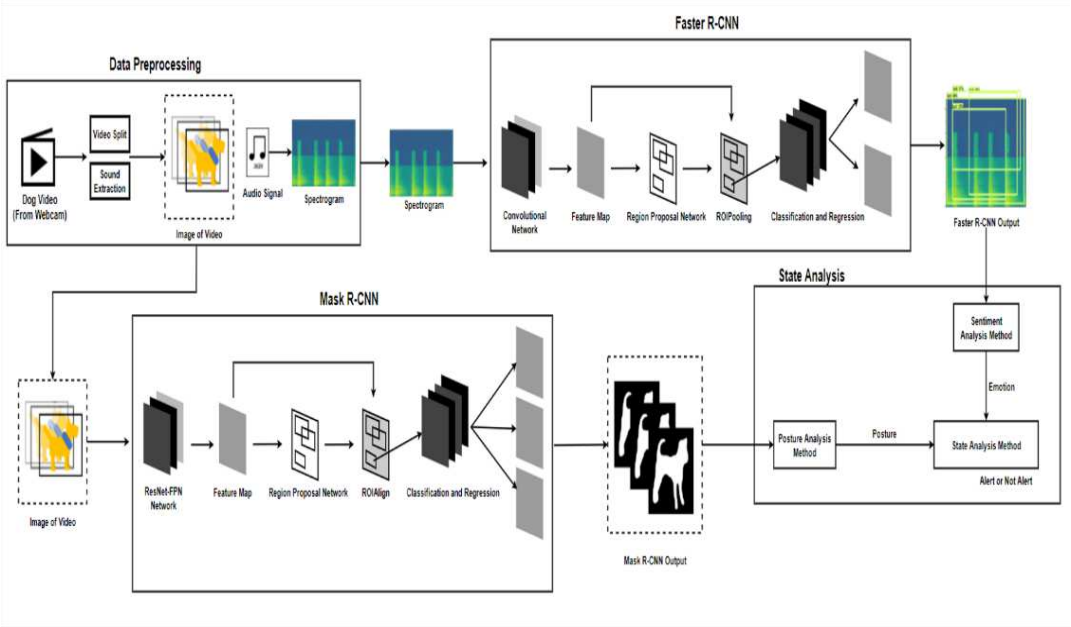

Figure 4: Identification Network Architecture

The the pet emotion analysis process of the smart Internet of Things system is shown in Figure 3, which includes three parts: the user side, the hardware side, and the software side. The pet body on the user side is the target of the analysed emotional state, and the owner's smart handheld device is the carrier for receiving notifications of pet state analysis. The hardware side is a smart webcam and a computing core platform. The smart webcam is used to capture pet video and audio files, and the pet video and audio information is input to the computing core platform for analysis, identification and notification. The software side is the environment and package tools of the computing core platform. Pet audio files are extracted and stored through MoviePy, and sound analysis pre-processing is performed through PyAudio, the Tensorflow deep

learning image recognition framework is used for emotion analysis and recognition, and then Mask R-CNN is used for mask generation and OpenCV-Python for pose analysis and recognition. The above emotion and posture analysis recognition results are used to determine the specific behaviour state of the pet. The specific behaviour state result is stored in the database and the owner is notified by the email package PyEmail for subsequent processing. The pet sentiment analysis network architecture of the smart Internet of Things system is shown in Figure 4, including data preprocessing, the Faster R-CNN neural network [15], Mask R-CNN neural network, and specific behaviour state analysis. The data preprocessing part performs image framing for the images recorded by the webcam, as well as having the function of extracting sound files and generating spectrograms. After dividing the image into frames, the Mask R-CNN neural network is performed to generate the contour mask map and the pose analysis algorithm is used to obtain the pose analysis result. The spectrogram uses Faster R-CNN neural network image recognition to obtain the sentiment analysis results. According to the above-mentioned posture and emotion analysis results, the pet's specific behaviour state is determined.

2.3 System Functions

```
1.  INPUT video
2.  Extract the video to the sound file
3.  Cut the movie to extract the image
4.  IF Image has dogs THEN
5.      Convert sound files into spectrograms
6.      Judging emotions by the voice of dogs
7.      Convert images into mask images
8.      Determine the dog's posture
9.      IF Emotion and posture reach a specific relationship THEN
10.         Send notification messages to users
11.     IFEND
12. ELSE
13.     Load new video
14. ENDIF
```

Figure 5: Main Function of System

The main functional flow of the system is shown in Figure 5. It uses a web camera to capture the video and audio of the pet body, and the core computing platform performs framed image and sound files to analyse the pet's posture and emotional information. When the pet's emotional analysis indicates a

specific state such as alert, the owner will be notified for processing. The system randomly samples the framed images of the pet body for Mask R-CNN object detection and obtains the contour mask map, then uses the posture analysis algorithm to obtain the pet posture information. The system converts sound files into spectrograms and uses Faster R-CNN for emotion recognition to obtain pet emotion information. After the system has successfully obtained the pet's posture and emotion information, it makes specific state association judgments to notify the owner of subsequent processing.

A. Mask R-CNN contour mask

The system is based on Mask R-CNN object detection to identify pets and generate contour masks. The sample set of contour masks in Figure 6 includes posture categories such as pets standing, sitting, and lying. The system sets the label category as background and pet. Two types are used for deep learning recognition model training to generate weight files for contour mask recognition.

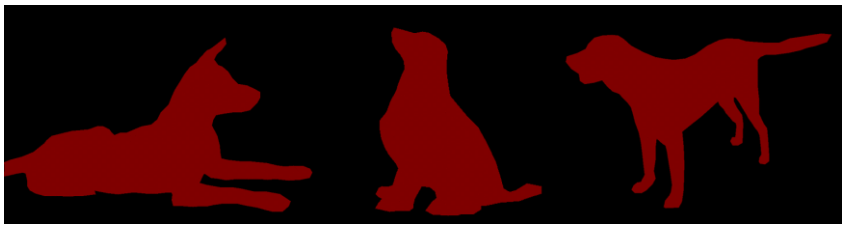

Figure 6: Pose Category of Contour Mask Sample Set

B. Faster R-CNN identification analysis

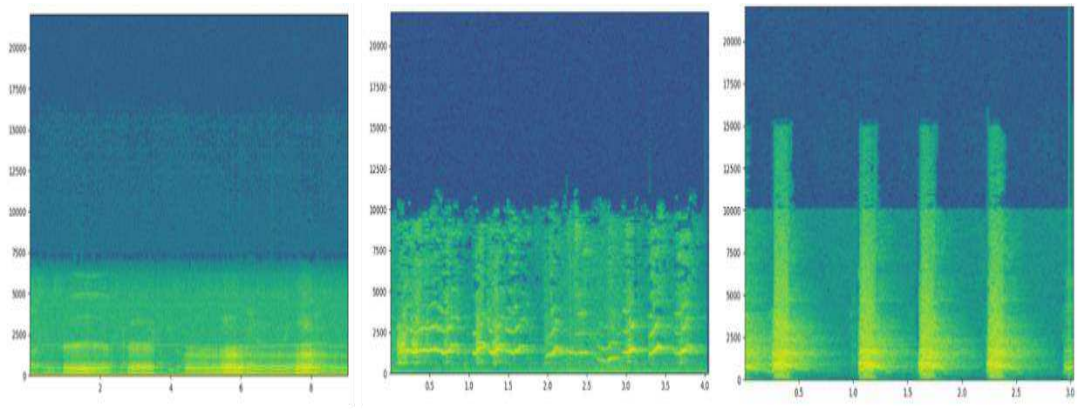

Figure 7: Spectrogram of Angry, Sad and Normal Mood Barking

Figure 7 shows spectrograms for the pet’s barking in different moods. The left picture is the angry spectrogram, the middle picture is the sad spectrogram, and the right picture is the normal barking spectrogram. The system is based on the Faster R-CNN network architecture to recognise the spectrogram of the

pet's emotional bark, and uses the deep learning recognition model to train and generate the weight file for emotion recognition.

2.4 System Algorithm

A. Posture analysis algorithm

The system posture analysis algorithm is shown in Figure 8. It performs image framing for the recorded video, and randomly selects β pieces of framed images as posture analysis samples, where β is less than or equal to the number of frames. When the selected framed image is judged to have a pet based on Mask R-CNN, the pose is judged to be empty; otherwise, a pet contour mask map of the framed image will be generated. The position of the pet in the image is found by using the contour mask. We use (x_{min}, y_{min}) to represent the coordinate value of the upper left corner of the object box, and use (x_{max}, y_{max}) to represent the coordinate value of the lower right corner of the object box. According to formula (1), we calculate the row position value of the most pet-rich area (white value) in the object frame and set it to the max_x value. According to formula (2), we calculate the position value of the column with the most pet areas (white value) in the object box and set it to the max_y value. We use IMG to represent the contour mask image array. The white value is 255 and the black value is 0.

$$max_x := \max_j \left(\sum_{j=x_{min}}^{x_{max}} img_{i=[y_{min}, y_{min}+1, \ldots, y_{max}], j} \right) \quad (1)$$

$$max_y := \max_i \left(\sum_{i=y_{min}}^{y_{max}} img_{i, j=[x_{min}, x_{min}+1, \ldots, x_{max}]} \right) \quad (2)$$

$$\|max_x - x_{min}\| \leq \|max_x - x_{max}\| \quad (3)$$

$$\|max_y - y_{min}\| \ / \ \|max_y - y_{max}\| \leq \alpha \quad (4)$$

$$\frac{\frac{\sum_{i=max_y}^{y_{max}} \sum_{j=x_{min}}^{max_x} img_{i,j}}{255}}{|max_x - x_{min}| \times |max_y - y_{max}|} \geq \kappa \quad (5)$$

$$\frac{\frac{\sum_{i=max_y}^{y_{max}} \sum_{j=max_x}^{x_{max}} img_{i,j}}{255}}{|max_x - x_{max}| \times |max_y - y_{max}|} \geq \kappa \quad (6)$$

According to the above values of x_{min}、y_{min}、x_{max}、y_{max}、max_x、$max_y \in Z^+$, the head direction of the pet is judged by the distance between the object's head and the left and right borders of the object. If the condition of formula (3) is met, the head of the pet in the framed image faces to the left;

otherwise, it faces to the right. The posture is judged by judging the distance between the head of the object and the upper and lower boundaries of the object. If the framed image with the pet's head facing to the left meets the condition of formula (4), it is judged to be standing. If the standing posture is not met, the ratio of the pet area to the background area of the framed image is judged. If the condition of formula (5) is met, it is judged as the prone posture, otherwise, it is the sitting posture. The framed image with the pet's head facing to the right still uses formula (4) to determine the standing posture conditions. If the standing posture is not met, formula (6) is used to determine the lying posture conditions. If none of them meets formula (4) and formula (6), the condition is sitting. This research paper designs $\alpha \ and \ \kappa \ \in \ R^{+}$ and sets α to be a real number of 1.2 and κ to a real number of 0.38.

```
INPUT  β images
INPUT  posture[β]
FOR  i=0  to  β-1 DO
  IF Image has dogs THEN
    Generate the contour mask of the i-th image
    Find the position of the i-th image dog
    IF The head of the dog in the i-th image is facing left THEN
      Analyze the posture of the dog in the i-th image
      IF Posture is standing THEN
        posture[i] set to standing posture
      ELSE IF Posture is lying THEN
        posture[i] set to lying posture
      ELSE
        posture[i] set to sitting posture
      ENDIF
    ELSE
      Analyze the posture of the dog in the i-th image
      IF Posture is standing THEN
        posture[i] set to standing posture
      ELSE IF Posture is lying THEN
        posture[i] set to lying posture
      ELSE
        posture[i] set to sitting posture
      ENDIF
    ENDIF
  ELSE
    posture[i] set to NULL
  ENDIF
ENDFOR
Determine the most posture in posture[ β ] as the posture information of the
video
```

Figure 8: Posture Analyse Algorithm

B. Sentiment analysis algorithm

The system sentiment analysis algorithm is shown in Figure 9. It extracts sound information from video files for sentiment analysis. Using the spectrogram as a sentiment analysis feature, the horizontal coordinate of the spectrogram is time, the vertical coordinate is frequency, and the coordinate point value is the speech data energy, as shown in Figure 7. The system defines sentiment analysis categories as angry, sad, and normal, and is based on the Faster R-CNN network architecture to train and recognise the spectrogram model. Faster R-CNN arranges the identified possibility results from high to low reliability to form a one-dimensional array. From the identified one-dimensional array results, the top five emotion results are obtained as the voting results of the emotion analysis, and finally the emotion with the highest number of votes is used as the final emotion analysis result of the voice.

```
1.  INPUT video
2.  IF Image has pet THEN
3.      Extract the sound files in the video
4.      Convert audio message into image format
5.      Image analysis with image recognition
6.      Voting for emotional judgment
7.      IF The result of emotional voting is anger THEN
8.          Determine that the pet mood in the film is angry
9.      IF ELSE The result of emotional voting is sad THEN
10.         Determine that the pet mood in the film is sad
11.     ELSE
12.         Determine that the pet mood in the film is normal
13.     ENDIF
14. ENDIF
```

Figure 9: Sentiment Analysis Algorithm

C. Specific state analysis algorithm

```
1. INPUT Posture result
2. INPUT Emotional result
3. Infer the state of the dog based on the correlation between the result of the
   posture and the result of the emotion
4. IF The status result of the dog is the specified status THEN
5.     Notify users of analyzed information by E-Mail
6.     Perform the next canine status analysis
7. ELSE
8.     Perform the next canine status analysis
9. ENDIF
```

Figure 10: Specific State Analysed Algorithm

The system specific state analysis algorithm is shown in Figure 10, which makes association judgments based on the results of posture and emotion analysis. As shown in Figure 11, the system determines that the alert state is defined as a pet standing and making angry sounds. If the above specific status occurs, an email is sent to the owner's smart handheld device application as a reminder notification.

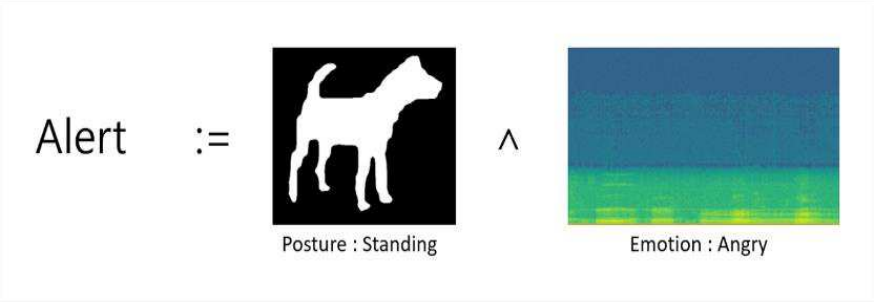

Figure 11: Definition of Alert Status

3. Evaluation and Results

3.1 Experimental platform and environment

Table I: Experimental Platform

Camera	Logitech Webcam C925
Platform	Computer
Programming Language	Python3.7
Main Library	Tensorflow == 1.14.0 Pycocotools == 2.0 Opencv-Python == 3.4.3.18 PyAudio == 0.2.11 Moviepy == 1.0.3 PyEmail

The experimental platform information is shown in Table I. It uses Logitech Webcam C925 as the network camera, the core computing platform is a embedded systems, the system is written using Python programming language with a library for the Tensorflow deep learning development environment, Pycocotools uses COCO library, OpenCV-Python image processing library, PyAudio speech processing library, MoviePy video editing library, and PyEmail email library.

3.2 Mask R-CNN mask training

The system implemented Mask R-CNN network architecture recognition model training with 475 pet images, and trained 60,000 steps to generate model weight files for contour mask recognition. The success rate of generating the

contour mask map with the training sample image set is 100% and the average cosine similarity accuracy is 96.78%. We add 10%, 30%, 50%, and 70% of the salt and pepper noise to the training sample image set to generate the contour mask. The respective success rates are 72.94%, 51.29%, 37.87% and 5.84%, and the average cosine is similar. The accuracy is 92.90%, 88.89%, 81.87% and 62.23% respectively. The values are shown in Table II. Figure 12 shows the similarity percentage data of the contour mask image generated by the standing, sitting and lying postures at different levels of noise. The similarity of the standing posture, sitting posture and lying posture without noise interference values is 96.45%, 97.01% and 97.27% respectively. The similarity of the standing, sitting and lying postures of the framed image under 10% noise interference is 92.63%, 93.81% and 93.00% respectively. The similarity of the standing, sitting and lying postures of the framed image under 30% noise interference is 89.98%, 87.73% and 87.77% respectively. The similarity of the standing, sitting and lying postures of the framed image under 50% noise interference is 85.60%, 74.12% and 85.68% respectively. The similarity of the standing, sitting and lying postures of the framed image under 70% noise interference is 69.81%, 60.33% and 67.06% respectively.

Table II: Mask Similarity

Added noise ratio	0%	10%	30%	50%	70%
Average Similarity	96.78%	92.90%	88.89%	81.87%	62.23%
Success Rate of Mask Generation	100%	72.94%	51.29%	37.87%	5.84%

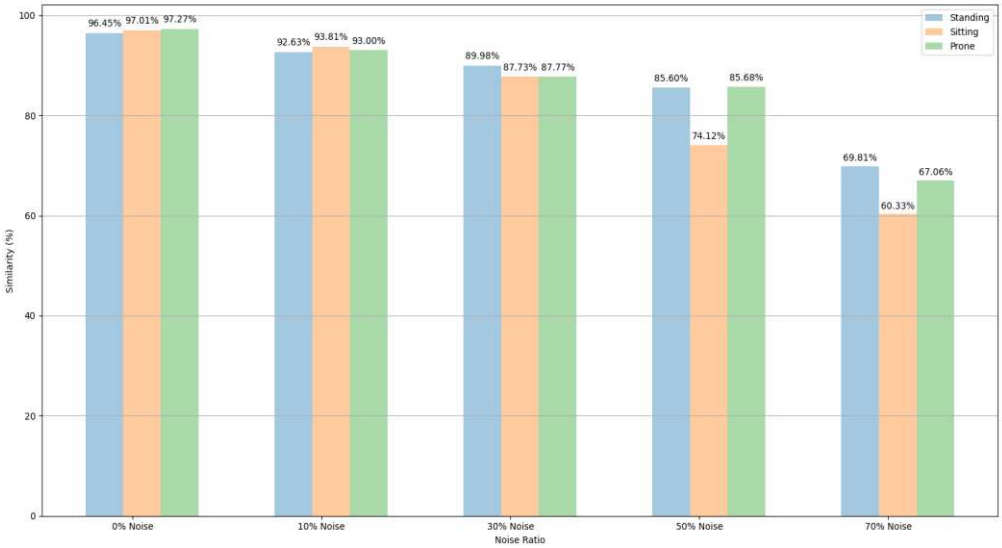

Figure 12: The Noise Level Corresponds to the Similarity of the Three Postures

3.3 Posture analysis

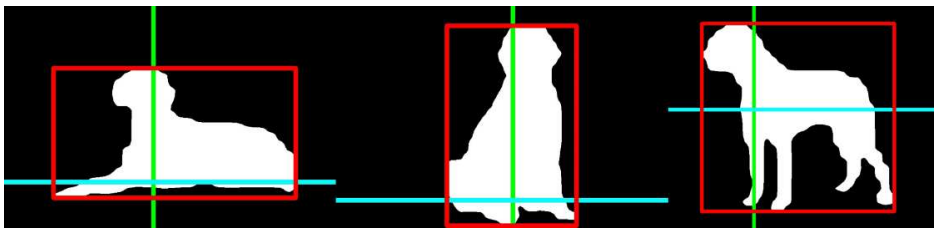

Figure 13: Posture Analysis Chart

The system is based on the contour mask map generated by Mask R-CNN to perform pose analysis. The results of the algorithm are shown in Figure 13 as lying, sitting, and standing. The red frame line is the target object position, the green line is the vertical position where the outline mask image contains the most target object information, and the cyan line is the horizontal position where the outline mask image contains the most target object information. The system uses 269 contour mask images for the pose analysis accuracy test and its accuracy is 90.7%.

3.4 Sentiment analysis

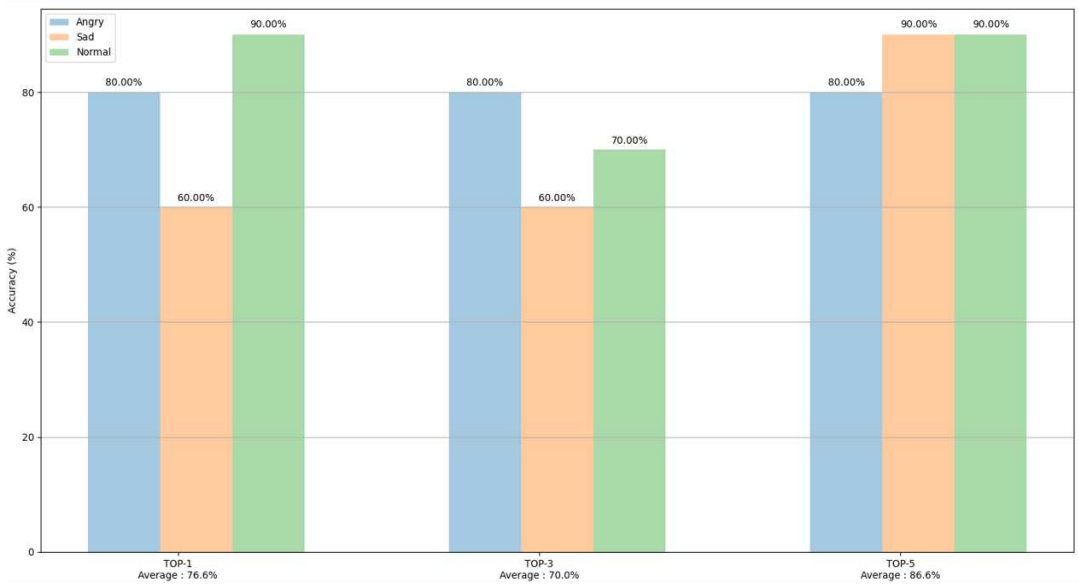

Figure 14: Sentiment Analysis Accuracy

TABLE III: MFCC + GMM-HMM Sentiment Analysis Accuracy

Emotion	Angry	Sad	Normal
Accuracy	10%	80%	10%
Average	33.33%		

The system uses 30 voice files to perform emotion analysis accuracy experiments, including three emotional states, angry, sad, and normal, and each emotional state has ten data. When the one-dimensional array identified by Faster R-CNN uses the TOP-1 result as the basis for emotion voting, the emotion with the highest vote is the final emotion result. The analysis accuracy of angry, sad, and normal states is 80%, 60%, and 90% respectively. The average accuracy is 76.6%, as shown in the histogram on the left of Figure 14. When the one-dimensional array identified by Faster R-CNN uses the TOP-3 result as the basis for emotion voting, the analysis accuracy of angry, sad, and normal states is 80%, 60%, and 70% respectively. The average accuracy is 70%. The value is shown in the middle histogram in Figure 14. When the one-dimensional array identified by Faster R-CNN uses the TOP-5 result as the basis for emotion voting, the analysis accuracy of angry, sad, and normal states are 80%, 90%, and 90% respectively. The average accuracy is 86.6%, as shown in the histogram on the right side of Figure 14. Using the TOP-5 results as the basis for emotion voting in the system, the average emotion accuracy is the best and the accuracy of each emotion is the most stable. In traditional voice recognition, MFCC plus GMM-HMM are used for voice recognition, and the same 30 voice files are used to perform emotion analysis accuracy experiments, including ten data for each of the three emotional states of angry, sad and normal. The accuracy of the analysis in angry, sad, and normal states is 10%, 80%, and 10% respectively. The average accuracy is 33.3%. The values are shown in Table III.

3.5 State recognition

Figure 15: Notification

When the system uses the pet specific state analysis algorithm to determine

that it is alert, it will immediately send an email to notify the owner, as shown in Figure 15. The system uses seven audiovisual files to determine the alert state and its accuracy is 85.71%. If the sentiment analysis uses MFCC plus GMM-HMM for voice recognition, the accuracy of judging the alert state is 0%.

4. Conclusions

This research paper proposes pet emotion analysis based on the smart Internet of Things system, using Mask R-CNN for pet object detection and contour mask generation, and Faster R-CNN for pet emotion recognition. The fusion of object posture and emotional characteristics is used as the basis for identification and analysis of pet emotions. Taking the detection of pets in an alert state as an example, the traditional deep learning image recognition technology cannot detect the above states. This research paper has an accuracy of 85.71% of successful identification in a non-contact manner and informs the owner for processing.

Declarations

Funding

This research was funded by National United University, Taiwan.

Conflict of interest

The authors declare that they have no conflict of interest.

Ethical standard

This article does not contain any studies with human participants or animals performed by any of the authors.

Author Contributions

Supervision, Ming-Fong Tsai; Writing – original draft, Ming-Fong Tsai and Jhao-Yang Huang; All authors have read and agreed to the published version of the manuscript.

References

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

[2] C. Ittichaichareon, S. Suksri and T. Yingthawornsuk, Speech Recognition using MFCC, International Conference on Computer Graphics, Simulation

and Modeling, pp. 135-138, 2012.
[3] M. Tsai, P. Lin, Z. Huang and C. Lin, Multiple Feature Dependency Detection for Deep Learning Technology - Smart Pet Surveillance System Implementation, Electronics, Volume 9, Issue 9, pp. 1387-1403, 2020.
[4] M. Hasan, M. Jamil, G. Rabbani and M. Rashidul, Speaker Identification using Mel Frequency Cepstral Coefficients, International Conference on Electrical & Computer Engineering, pp. 565-568, 2004.
[5] M. Hunt, M. Lennig and P. Mermelstein, Experiments in Syllable-based Recognition of Continuous Speech, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 880-883, 1980.
[6] M. Gales and S. Yang, The Application of Hidden Markov Models in Speech Recognition, Foundations and Trends in Signal Processing, vol. 1, no. 3, pp. 195-304, 2007.
[7] L. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, 1989.
[8] B. Schuller, G. Rigoll and M. Lang, Hidden Markov Model-based Speech Emotion Recognition, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 401-404, 2003.
[9] L. Muda, M. Begam and I. Elamvazuthi, Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques, Journal of Computing, vol. 2, no. 3, pp. 138-143, 2010.
[10] C. Chen and A. Bilmes, MVA Processing of Speech Features, IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, no. 1, 257-270, 2007.
[11] A. Benba, J. Abdelilah and A. Hammouch, Discriminating between Patients with Parkinson's and Neurological Diseases using Cepstral Analysis, IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 24, no. 10, pp. 1100-1108, 2016.
[12] N. Dave, Feature Extraction Methods LPC, PLP and MFCC in Speech Recognition, International Journal for Advance Research in Engineering and Technology, vol. 1, no. 6, pp. 1-5, 2013.
[13] E. Brigham, The Fast Fourier Transform and Its Applications, Prentice Hall, Englewood Cliffs, New Jersey, 1988.
[14] N. Ahmed, T. Nasir and K. Rao, Discrete Cosine Transform, IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90-93, 1974.
[15] 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.

Figures

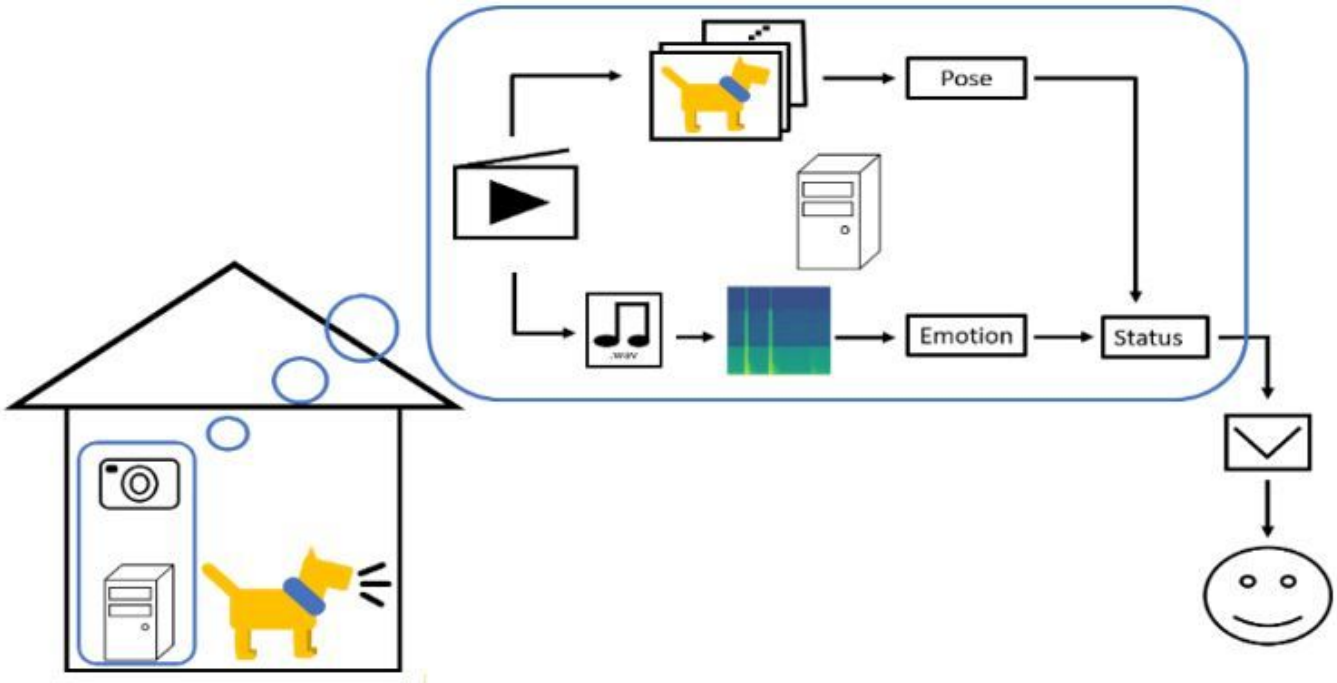

Figure 1

System Overview

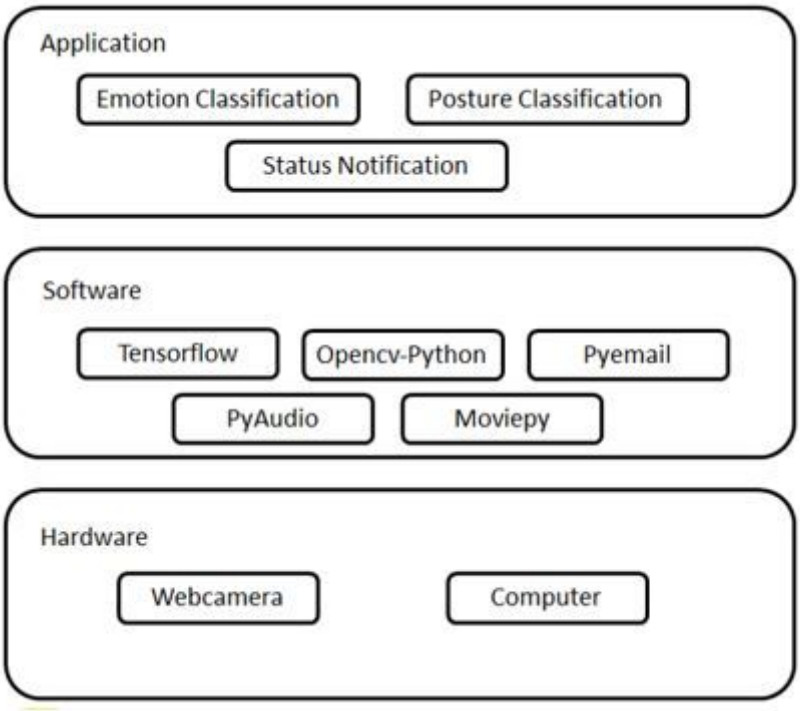

Figure 2

System Architecture Diagram

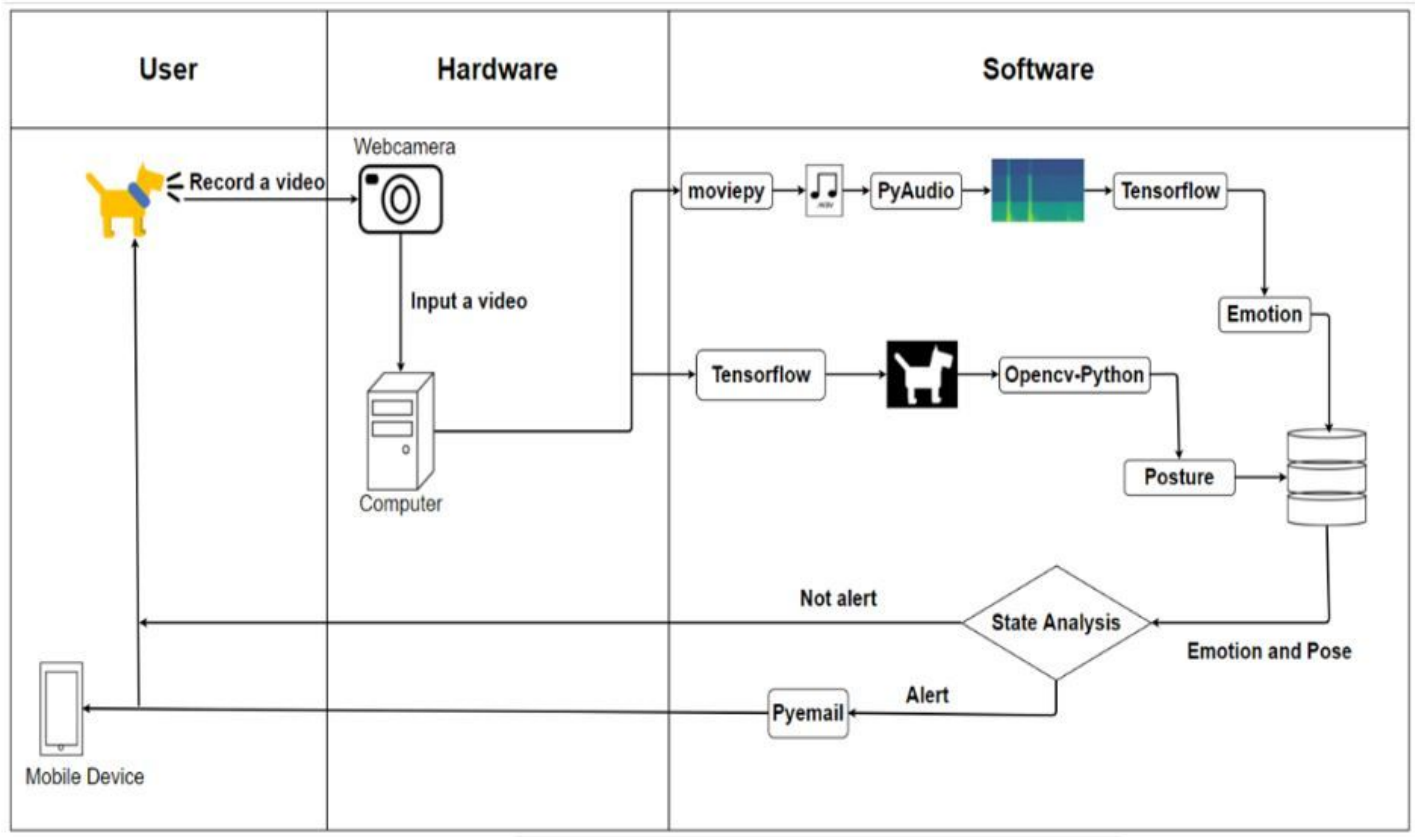

Figure 3

System Fowchart

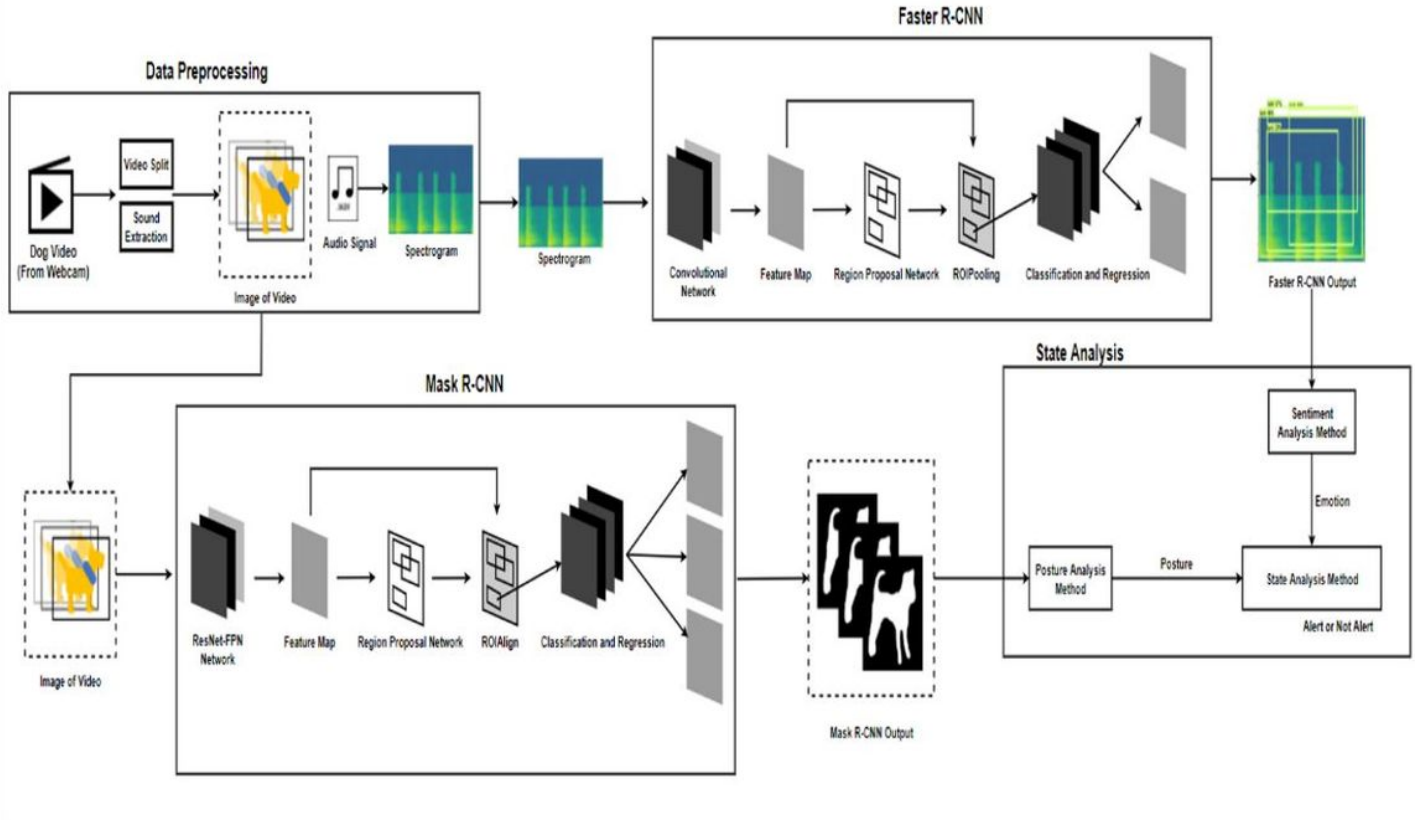

Figure 4

Identification Network Architecture

```
1.  INPUT video
2.  Extract the video to the sound file
3.  Cut the movie to extract the image
4.  IF Image has dogs THEN
5.      Convert sound files into spectrograms
6.      Judging emotions by the voice of dogs
7.      Convert images into mask images
8.      Determine the dog's posture
9.      IF Emotion and posture reach a specific relationship THEN
10.         Send notification messages to users
11.     IFEND
12. ELSE
13.     Load new video
14. ENDIF
```

Figure 5

Main Function of System

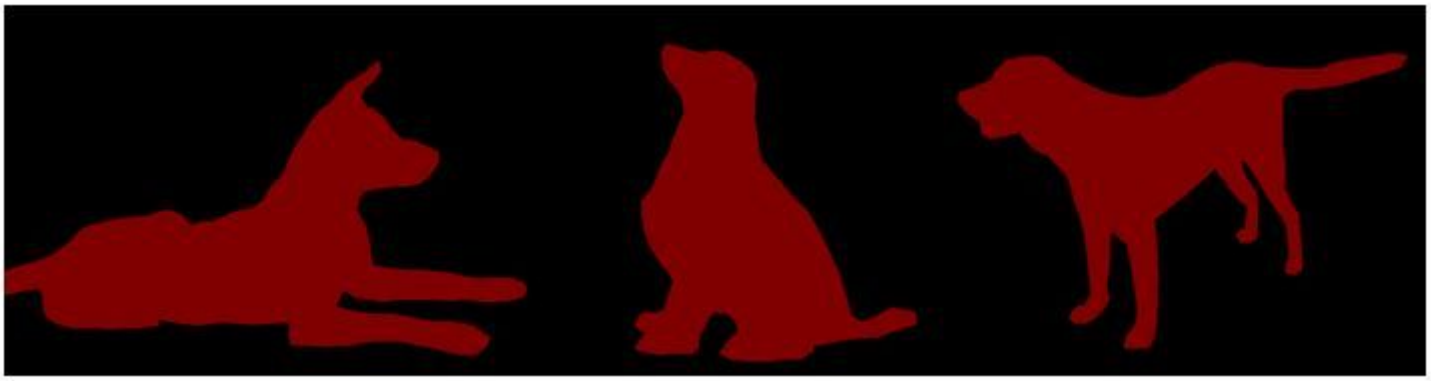

Figure 6

Pose Category of Contour Mask Sample Set

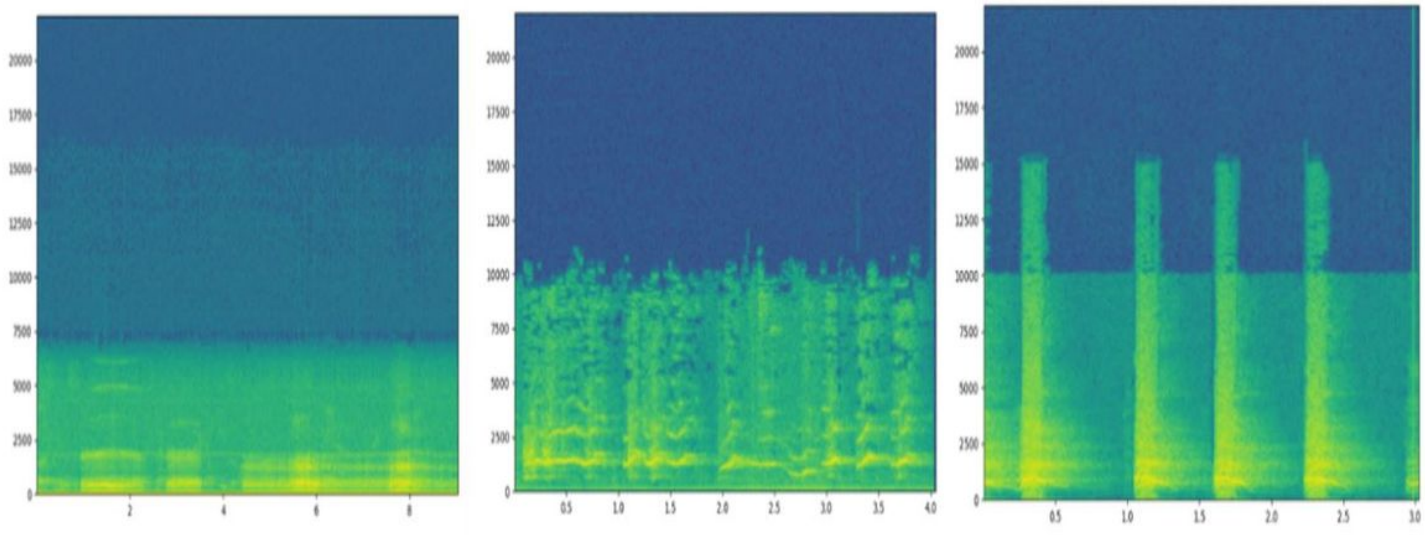

Figure 7

Spectrogram of Angry, Sad and Normal Mood Barking

```
1.  INPUT  β images
2.  INPUT  posture[β]
3.  FOR  i=0  to  β-1 DO
4.    IF Image has dogs THEN
5.      Generate the contour mask of the i-th image
6.      Find the position of the i-th image dog
7.      IF The head of the dog in the i-th image is facing left THEN
8.        Analyze the posture of the dog in the i-th image
9.        IF Posture is standing THEN
10.         posture[i] set to standing posture
11.       ELSE IF Posture is lying THEN
12.         posture[i] set to lying posture
13.       ELSE
14.         posture[i] set to sitting posture
15.       ENDIF
16.     ELSE
17.       Analyze the posture of the dog in the i-th image
18.       IF Posture is standing THEN
19.         posture[i] set to standing posture
20.       ELSE IF Posture is lying THEN
21.         posture[i] set to lying posture
22.       ELSE
23.         posture[i] set to sitting posture
24.       ENDIF
25.     ENDIF
26.   ELSE
27.     posture[i] set to NULL
28.   ENDIF
29. ENDFOR
30. Determine the most posture in posture[ β ] as the posture information of the
    video
```

Figure 8

Posture Analyse Algorithm

```
1.  INPUT video
2.  IF Image has pet THEN
3.      Extract the sound files in the video
4.      Convert audio message into image format
5.      Image analysis with image recognition
6.      Voting for emotional judgment
7.      IF The result of emotional voting is anger THEN
8.          Determine that the pet mood in the film is angry
9.      IF ELSE The result of emotional voting is sad THEN
10.         Determine that the pet mood in the film is sad
11.     ELSE
12.         Determine that the pet mood in the film is normal
13.     ENDIF
14. ENDIF
```

Figure 9

Sentiment Analysis Algorithm

```
1.  INPUT Posture result
2.  INPUT Emotional result
3.  Infer the state of the dog based on the correlation between the result of the
    posture and the result of the emotion
4.  IF The status result of the dog is the specified status THEN
5.      Notify users of analyzed information by E-Mail
6.      Perform the next canine status analysis
7.  ELSE
8.      Perform the next canine status analysis
9.  ENDIF
```

Figure 10

Specific State Analysed Algorithm

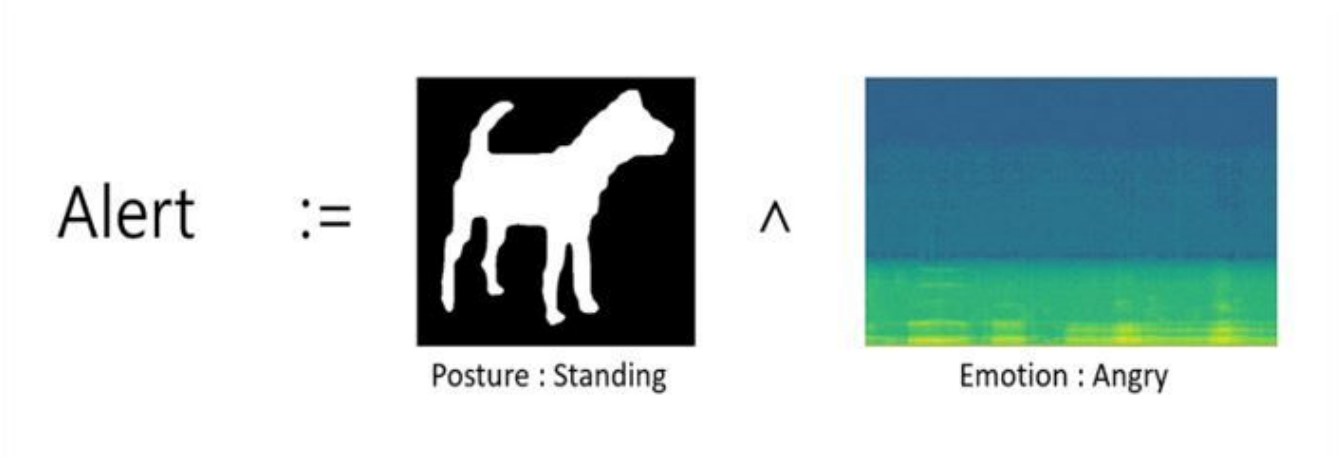

Figure 11

Definition of Alert Status

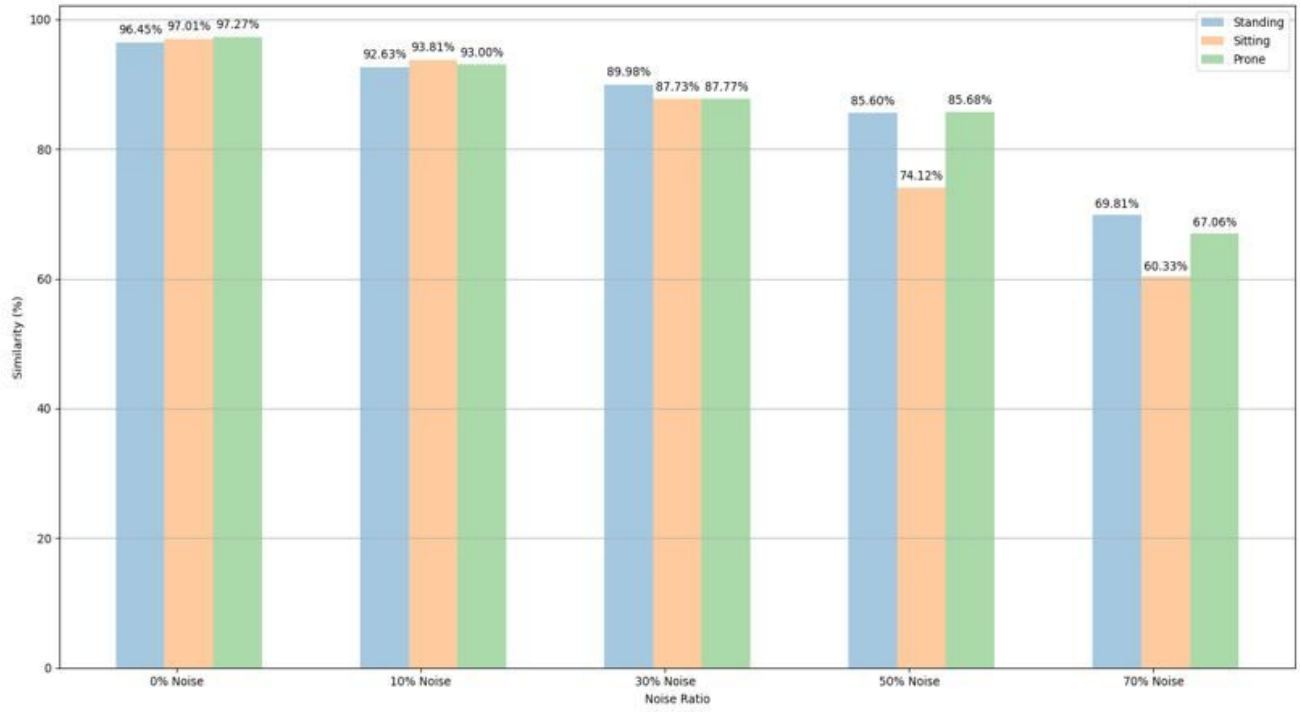

Figure 12

The Noise Level Corresponds to the Similarity of the Three Postures

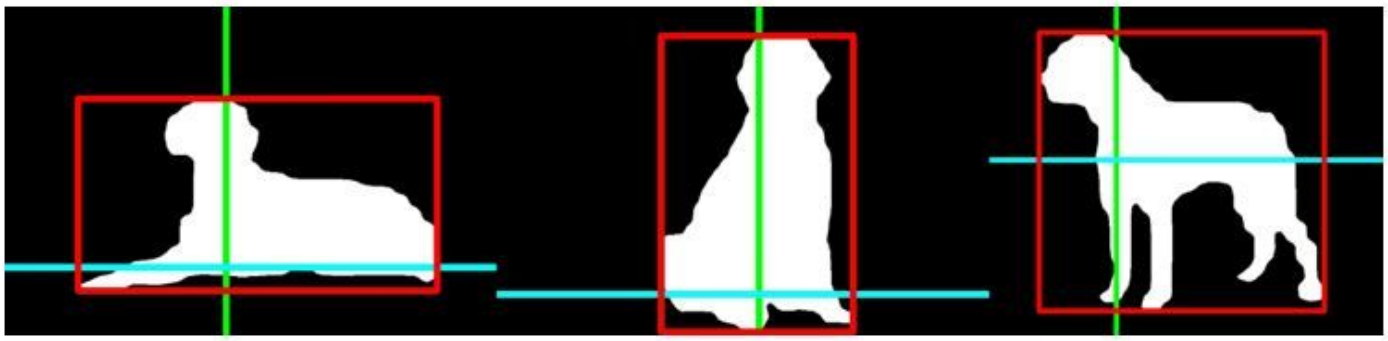

Figure 13

Posture Analysis Chart

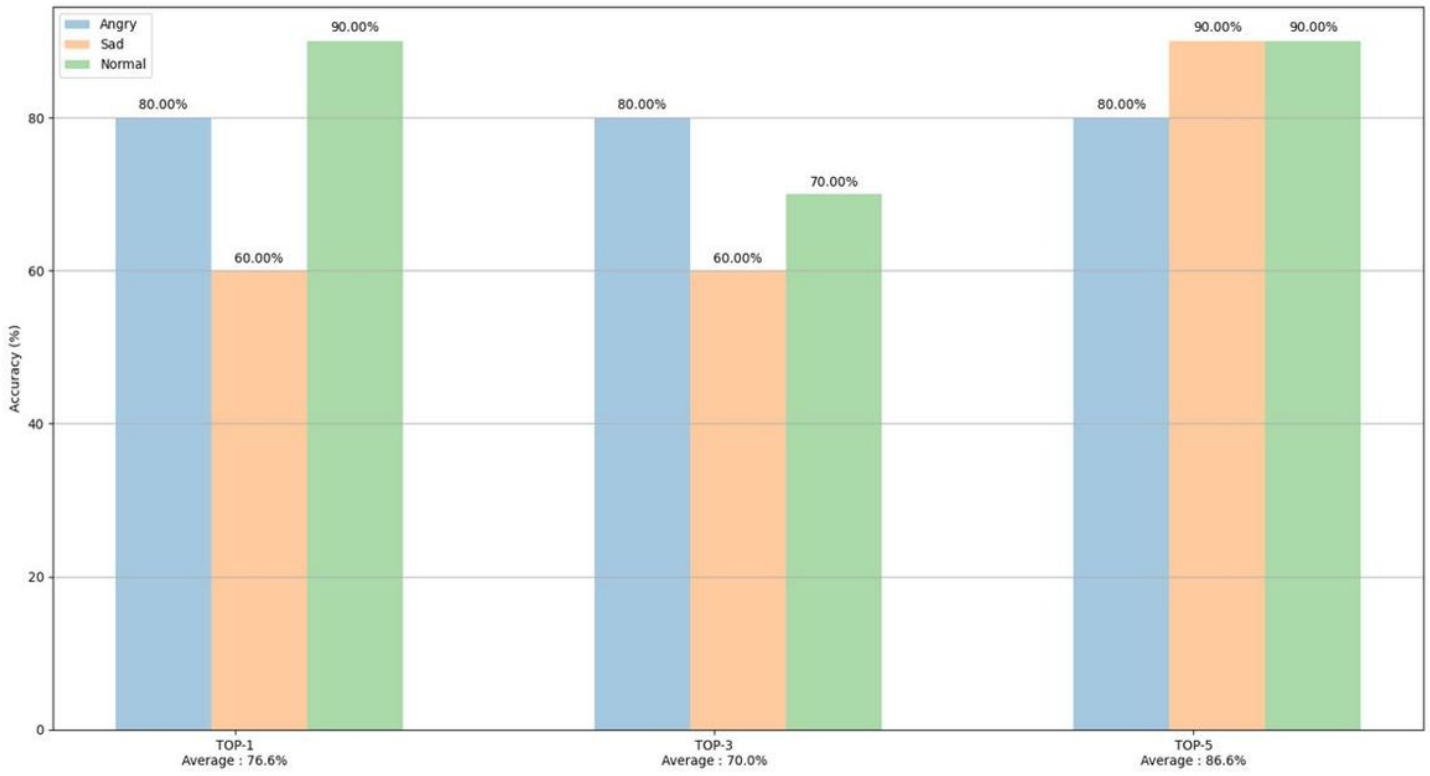

Figure 14

Sentiment Analysis Accuracy

Figure 15

Notification