Canine Behavior Interpretation Framework Using Deep Graph Model

Jongmin Lim(✉), Donghee Kim, and Kwangsu Kim

Sungkyunkwan University, Suwon, Korea
jm.lim@g.skku.edu, {ym.dhkim,kim.kwangsu}@skku.edu

Abstract. Humans have long aspired to understand dog behavior. While research on the Calming signal has achieved substantial progress in understanding dog behavior, it remains an unfamiliar concept to non-expertise. Therefore, in this paper, we introduce a framework for analyzing dog behavior, which defines the interrelationship between dog postures through a graph model without any additional devices but a camera. First of all, our framework classifies the dog posture in frame units, using a machine learning model based on the position information of the dog's body part in the video captured by the camera. We then analyze dog behavior using graph models that define interrelationships among classified dog postures. We expect that our approach will help non-expertise to understand dog behavior.

Keywords: Canine behavior analysis · Object detection · Graph model

1 Introduction

For centuries, dogs have socially interacted with humans by playing various roles such as hunters, security guards, and friends. As such, dogs have come to be thought of as spiritual companions rather than mere possessions for one's pleasure. Nevertheless, humans and dogs have many differences. In particular, unlike a human, dogs communicate non-verbally using body language. To interpret their behavior is more difficult for humans because of these differences. Therefore, keeping dogs would become handy if it is possible to understand dog's behavior.

Norwegian dog trainer Turid Rugaas defines at least 30 types of "Calming signals [6]" to understand dog behavior. The calming signal is a communication method with dogs each other, but also known to use these signals with humans

This research was partly supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2021-2015-0-00742) and Institute of Information & communications Technology Planning & Evaluation (IITP) (No. 2020-0-00990, Platform Development and Proof of High Trust & Low Latency Processing for Heterogeneous·Atypical·Large Scaled Data in 5G-IoT Environment).

© Springer Nature Switzerland AG 2021
L. Rutkowski et al. (Eds.): ICAISC 2021, LNAI 12854, pp. 99–110, 2021.
https://doi.org/10.1007/978-3-030-87986-0_9

too. Research on the calming signal has helped many dog trainers and behaviorists and has led to the development of dog ethology. However, it is still arduous for non-expertise to understand dog ethology without any knowledge of biology, genetics, evolution, etc.

Moreover, various studies have been done using high-quality equipment, yet attaching the equipment might cause stress for dogs, and it is too price for commercializing.

Also, Previous studies have been defined dog behaviors by analyzing a single posture. But it neglects the continuity of dog postures. For example, a dog's tail-raising posture is usually considered a friendly signal. Nevertheless, the tail-raising posture could sign fear, anxiety, danger, or warning depending on the previous postures.

Therefore, in this paper, we will introduce a framework for dog behavior analysis by defining the interrelationship between dog postures, using a graph model without any additional devices but a camera. The graph model uses various dog postures as nodes and defines the continuous interrelationships between nodes in reference to the calming signal. First, Our framework uses object detection to detect dogs and their body parts from the video captured by the camera. Then, the detected dog area is set as absolute coordinate space to utilize the location information of each body part detected regardless of the size and position of the dog in the image. And, a dog's posture is classified in frame units through a machine learning model based on the absolute coordinate values of the body parts. Finally, the graph model determines the dogs' behavior using the postures classified on frame units. Our framework is extensible to adding a new dog posture as a node and redefining the continuous interrelationships between nodes.

Organization. In Sect. 2, We discuss about basic knowledge on dog behaviors. In Sect. 3, we explain our posture classification method and graph model. In Sect. 4, we evaluate the training process and performance of dog posture classification models. Lastly, we conclude the study in Sect. 5.

2 Dog Behaviors

A dog's body language is a sophisticated non-verbal system that non-understand in a single posture. A single posture is only part of the package that displays its mood. Therefore, skilled dog trainers and behaviorists do not analyze a dog's behavior with just a single posture but observe every posture expressed in succession. For example, a beginner trainer may believe that when a dog lifts one paw, it is emotional such as anxiety, fear, or stress, or describe as being hurt. However, experienced trainers take into account the previous postures the dog has shown. If a dog raises its paw after tilting its face, it can express curiosity and expectation. Also, if the dog raises its paws after raising its tail, this may be an action to get someone's attention. Therefore, it is imperative to consider all the postures they have shown when decoding dog behavior. In the next chapter, we explain how a dog's behavior can be analyzed through dog posture continuity (Fig. 1).

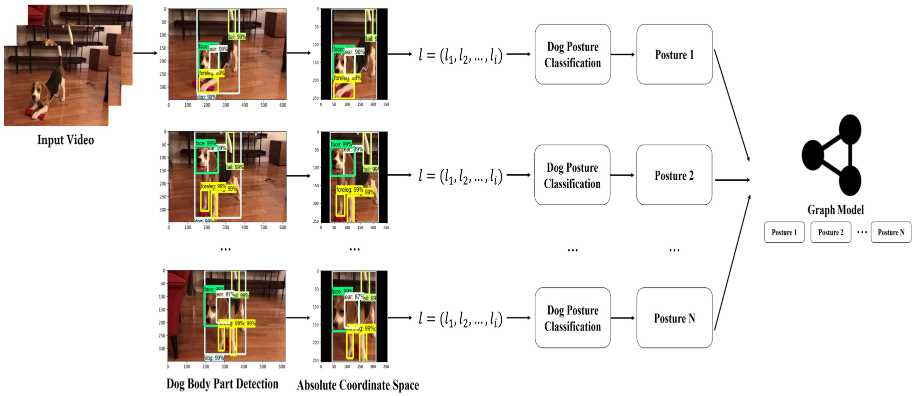


Fig. 1. Overall process of Canine Behavior Interpretation Framework. Given a video, detects the dog's body part in a frame unit. Each body part's location information is extracted after setting the detected dog area as an absolute coordinate space. Then, the dog posture is classified through a machine learning algorithm based on the extracted body part location information. Later, dog behavior is analyzed through a graph model that defines postures' continuous interrelationships classified in frame units.

3 Method

We introduce a graph model that sequentially represents an interrelationship of a dog's postures. This graph model puts the nodes defined in various dog postures and explores the successive interrelationships between each node to understand dog behavior. To implement this model, we first detect dogs and their body parts using the object detection method [4,5] in the video. We then project the detected dog area as absolute coordinate space and classify the dog's posture in frame units using each body part's coordinates. After that, the graph model determines the dogs' behavior using the posture classified in frame units.

3.1 Dog Body Parts Detection Using Object Detection

It is vital to observe the dog's body parts' location to understand its body language. In previous studies, researchers attached sensors to dog body parts for movement analysis. However, dogs were reluctant to have the sensors attached.

Therefore, we established our goal to detect the dog's body part through a deep learning-based object detection method without any additional equipment but a camera to extract the dog's body part's location information. For this goal, we collected an image dataset of beagles among dog breeds through YouTube. Yet, considering that the deep learning-based object detection method requires a vast dataset, we encountered a problem that it is too expensive to collect and label additional datasets.

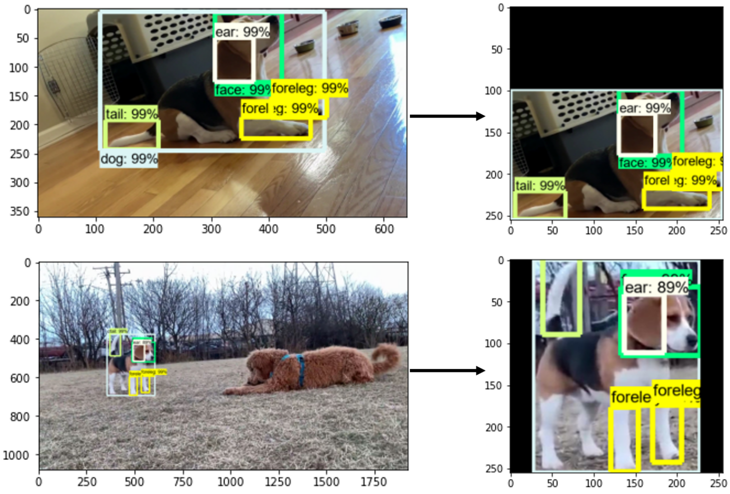


Fig. 2. We set the absolute coordinate space to utilize the dog's body part's position information without space constraints. Black is padding to maintain spatial information of the dog's body part.

To address this problem, we decided to use transfer learning [7]. Transfer learning is a training method that uses pre-trained models in similar domains to the corresponding model when data is deficient. The advantage of transfer learning use is previous learning experiences are adaptable for related tasks. Thus, a model is trained to detect dog body parts by applying transfer learning with a few samples, and as a result, it is possible to get the location information of dog and body parts from the image without using expensive equipment.

3.2 Dog Posture Classification Using Machine Learning

Once the dog and its body parts are detected from the video, the next step is to classify the dog's postures using the detected body parts' coordinates. Setting an absolute coordinate space is necessary to accomplish the exact coordinate. As shown in Fig. 2, the absolute coordinate space allows us to obtain fixed position information of body parts regardless of the size and position of the dog in the image. Our absolute coordinate space is established in the following process. First, To maintain the spatial information of the detected dog area, padding is added its area according to the ratio of width and height. We then resize the dog area where the padding was added in the same size and set it as absolute coordinate space. Once the absolute coordinate space is set, the detected dog's body parts center coordinate values are extracted based on the absolute coordinate space. And the dog's postures are classified in frame units through a machine learning model based on the absolute coordinate values. Our dog posture classifier yields an expected percentile value.

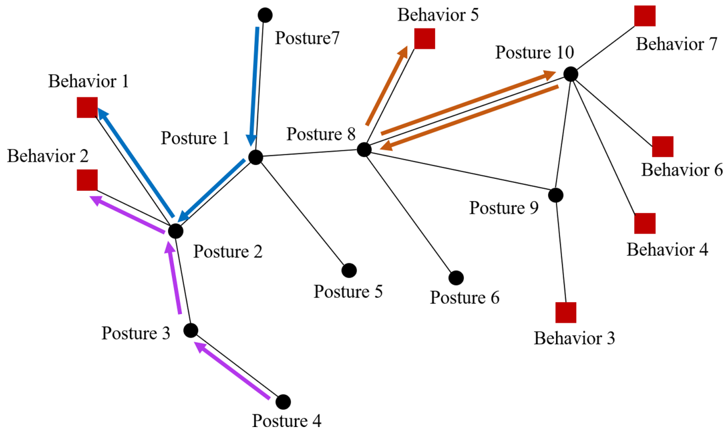


Fig. 3. We analyze dog behavior by considering posture change based on a graph model. For example, posture 2, which is finally classified, can be analyzed into two behaviors according to the postures classified in the previous frame. (Blue vs. purple). (Color figure online)

In this paper, following notation for the extracted absolute coordinate values. We define the *body part vector* as $\mathbf{l} = (\mathbf{l_1}, \mathbf{l_2}, .., \mathbf{l_i}), i \in \{1, 2..., k\}$, where $\mathbf{l_i}$ is the (x, y) absolute coordinate value of the k^{th} dog body parts.

3.3 Dog Behavior Graph Model Generation

Deep learning based object detection methods and machine learning are applied to classify dog postures in frame units. However, a single posture is merely part of the package that expresses a dog's mood. Therefore, we generated a graph model to analyze the behavior using the posture patterns that show continuously. The process of generating a graph model is as follows. First, combine successive postures and define them as behavior concerning the calming signal. After that, generate a graph that expresses the posture as nodes and the posture order as an edge. Also, the defined behavior is added as the end node of the graph, and the behavior node contains the information of the dog posture sequence. Finally, compose various graphs that define dog behavior through posture sequences into one graph model as shown in Fig. 3.

We denote for dog behavior set as $B = \{b_1, b_2, b_3..., b_n\}$, where b_n is a representation of variable length sequences data and o_i indicates one element of the dog posture set $P = \{p_1, p_2, .., p_n\}$.

$$b_n = (o_1 \rightarrow o_2 \rightarrow o_3 \rightarrow ... \rightarrow o_T)^T \tag{1}$$

Algorithm 1. Canine Behavior Interpretation Framework

Input: Video frames f
Output: Behavior
Dog Behavior set $B = \{b_1, b_2, b_3, ..., b_n\}$
prevposture ← Posture classified from previous frame
route ← graph travel route
for $f = 1$ to n **do**
  Obtain body parts vector $\mathbf{l} = \{\mathbf{l_1}, \mathbf{l_2}..., \mathbf{l_i}\}$
  posture ← PostureClassifier($\mathbf{l}$)
  if *posture* ∈ neighbors(*prevposture*) **then**
    route ←Insert(*posture*)
  end if
end for
if *route.endnode* is connected to behavior nodes **then**
  for behavior **in** connected behavior nodes **do**
    Choose higher-similarity behavior to the *route*
  end for
else
  for $b_j \in B$ **do**
    Chooses highest-similarity behavior to the *route*
  end for
end if
return behavior

3.4 Dog Behavior Analysis Using Graph Model

Once the graph model is built, our framework analyzes the dog's behavior as in Algorithm 1. First, approaches from the first frame to the n^{th}frame in the input video. And, the posture classifier expects the dog posture for each frame based on the *body part vector* $\mathbf{l}$ by setting a threshold with a probability of 0.8. With the first classified posture as the starting node, the graph model is explored by sequentially traveling to the next frame's classified posture node. However, incorrect posture classification can lead to an error in analyzing dog behavior. Therefore, if the classified posture from the current frame is not a neighbor node of the previous posture node in the graph model that defines the relationship between postures, it is considered misclassified and does not travel.

After exploring the graph up to the n^{th} frame of the input video with this method, our framework determined the dog behavior by dividing the information into 2 cases. If the end node is connected to the behavior node, measure the similarity between the travel route and posture sequence information of each connected behavior node and determine the behavior node with a higher similarity as the final behavior. And, if the similarity under the threshold is gained, or the end node is unconnected to the behavior node, the similarity is computed between the travel route and posture sequence information of all behavior nodes. Then, the dog behavior of the highest similarity value is considered as an output.

The similarity between the travel route and the posture sequence of the behavior node is measured through the graph editing distance [3] with the following Eq. 2.

$$d(g1, g2) = 1 - \frac{|mcs(g1, g2)|}{max(|g1|, |g2|)} \tag{2}$$

where, $d(g1, g2)$ means graph edit distance and $mcs(g1, g2)$ indicate the maximum common sub graph [1]. Also, $|g|$ indicate the size of graph. If the graph edit distance [3] is close, it means that the two graphs have high similarity.

4 Experiment

4.1 Evaluation for Dog Body Parts Detection

This section evaluates the object detection method's training process and performance to detect dog body parts.

Dataset and Traning. One of the most critical tasks is to prepare quality datasets when training the deep learning-based object detection method. To prepare a high-quality dataset of dog's body parts, we collected a total of 9,282 images of beagles with various backgrounds from YouTube. All images are annotated by drawing a bounding box on the dog and body part utilizing LabelImg [8] Then the images are converted to XML files in PASCAL VOC format [2]. Although high-quality datasets have been collected adequately through this process, an enormous amount of data is required to train the object detection method. However, acquiring a larger dataset is cost-prohibitive.

To overcome these barriers, we applied transfer learning [7]. A pre-trained Faster RCNN and SSD Mobilenet model is taken that uses a COCO dataset containing a large amount of animal data provided by the TensorFlow Model Zoo [9].

Result. We compared the performance of Faster R-CNN and SSD Mobilenet trained through transfer learning. We first set the IOU (Intersection over Union) threshold to compare the two models' mean average precision (mAP). The mAP is the result of precision and recalls "precision-recall" calculations on determining bounding boxes. Also, we observed the FPS(Frame per second), an important index in real-time detection. Table 1 shows mAP and FPS of Faster R-CNN and SSD Mobilenet. Faster R-CNN was an accurate model with high mAP but showed a poor FPS to detect dog's body parts in real-time. In contrast, SSD Mobilenet was effective in detecting dog body parts in real-time by showing good mAP and FPS.

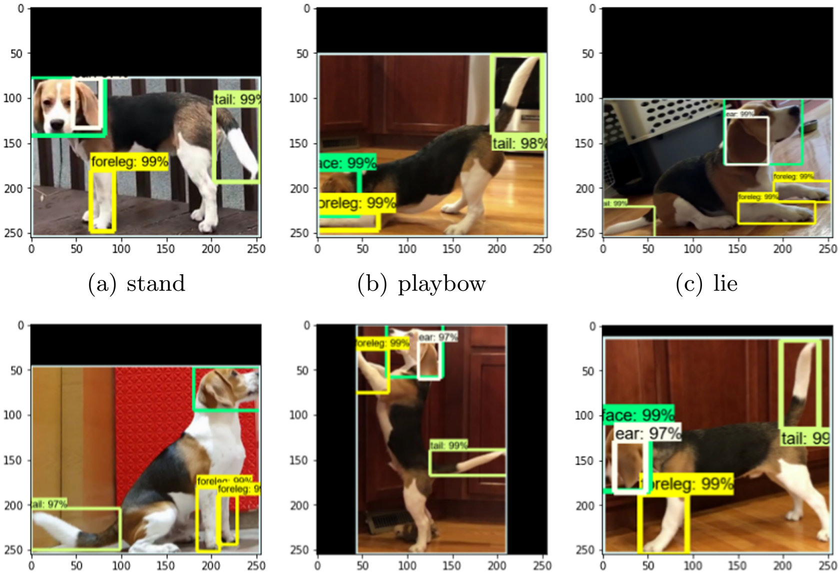
Table 1. Dog body part detection comparison

Method	mAP@0.50	mAP@0.75	FPS
Faster R-CNN	0.97	0.91	2
SSD Mobilenet	0.89	0.63	32

4.2 Evaluation for Dog Posture Classification

This section evaluates the machine learning model's training process and performance classified into six postures(stand, playbow, lie, sit, stand on two paws, tail raising) as shown in Fig. 4.

Dataset and Traning. Our dog's body part detection model emits the coordinate values of each body part. We constructed a dataset of dog posture classifiers by converting the emitted coordinate values into absolute coordinate values. Then, trained the Decision Tree, Neural Network, and Support Vector Machine among machine learning algorithms with the Configured dataset.



(a) stand (b) playbow (c) lie

(d) sit (e) stand on two paws (f) tail raising

Fig. 4. Example of classifying dog postures through machine learning based on the absolute coordinates of each body part.

Result. We first observed the accuracy of machine learning algorithms. As shown in Table 2, the Support Vector Machine's accuracy was the highest at 99.9%.

Also, we evaluated the Support Vector Machine's performance, the most accurate algorithm through the Confusion Matrix as shown in Fig. 5. A small misclassification occurred mainly between Tail raising and Playbow. This happens because the absolute coordinate values are similar between the two classes. As shown in Figs. 4(f) and 4(b), the difference between the two postures is in the position of the dog's upper body. However, the dog's upper body is in

Table 2. Dog posture classification compare

Algorithms	Accuracy
Decision Tree	85.6%
Neural Network	92.4%
Support Vector Machine	**99.9%**

an ambiguous position when analyzing the misclassified images. Although the postures classifier caused some misclassification, it is not a big problem in our framework for analyzing dog behavior by exploring a graph model that defines the interrelationships between postures.

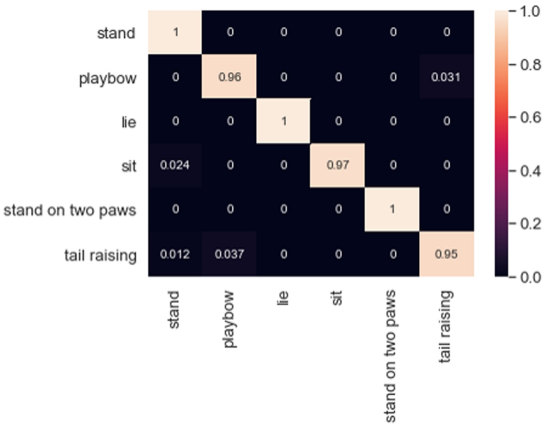


Fig. 5. Confusion matrix.

4.3 Experiment for Dog Behavior Analysis Using Graph Model

In this section, we describe the dog behavior defined according to the calming signal [6] and the experiment that analyzes the behavior for various videos based on the graph model.

Definition. As shown in Table 3, we defined the dog's behavior into seven types (Playful, Suspicious, Demanding, Relaxed, Peaceful, Interested, Joyful) according to the dog's posture change, and then generated a dog behavior graph model. However, according to dog trainers and behaviorists' needs, the specific behaviors are open to addition or modification.

Table 3. Defined dog behavior according to posture sequence

Posture sequence	Behavior
playbow→tail raising	Playful
tail raising→tail raising	Suspicious
tail raising→sit	Demanding
sit→sit	Relaxed
lie→lie	Peaceful
stand→tail raising→stand on two paws	Interested
sit→stand on two paws	Joyful

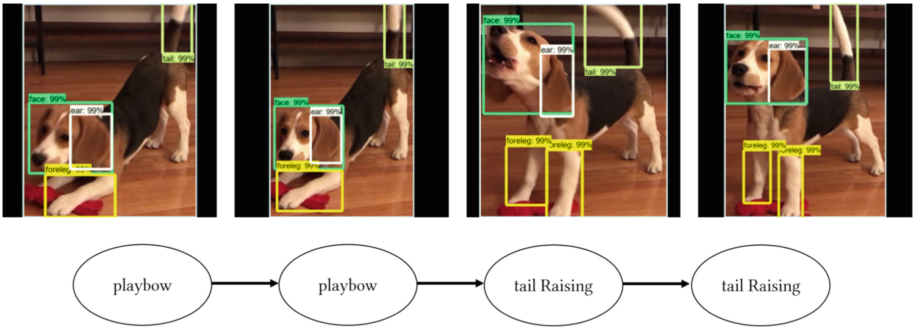
Table 4. Graph Edit Distance. The closer the graph edit distance is to 0, the higher the similarity between the two graphs. Conversely, as the graph edit distance is closer to 1, the similarity between the two graphs low.

	Video 1	Video 2	Video 3
Playful	**0.0**	**0.833**	1.0
Suspicious	0.666	**0.833**	1.0
Demanding	No calculation	0.833	0.75
Relaxed	No calculation	1.0	0.75
Peaceful	No calculation	1.0	1.0
Interested	No calculation	**0.166**	0.8
Joyful	No calculation	0.833	**0.25**

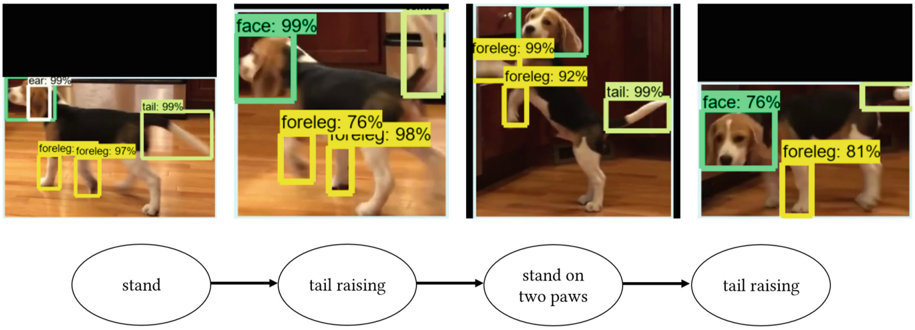
When the End Node is Connected to Behavior Nodes. Figure 6(a) shows the travel route(playbow→tail raising) created from Sample Video 1. The end node of the created travel route is connected to the behavior node Playful and Suspicious when according to Table 3. Our framework evaluates the similarity between each posture sequence information of connected behavior nodes(Playful and Suspicious) and the travel route to determine the final behavior between Playful and Suspicious. As a result, the behavior is regarded as Playful with higher similarity, as shown in Table 4.

However, as shown in Fig. 6(b), the end node of the travel route(stand→tail raising→stand on tow paws→tail raising) created from Sample video 2 is also a Tail raising, but when referring to Table 4, the similarity with the connected behavior node (Playful, Suspicious) is very low. Therefore, the similarity between the travel route and posture sequence of all the behavior nodes is measured, and the final behavior is determined as Interested with the highest similarity as shown in Table 4.

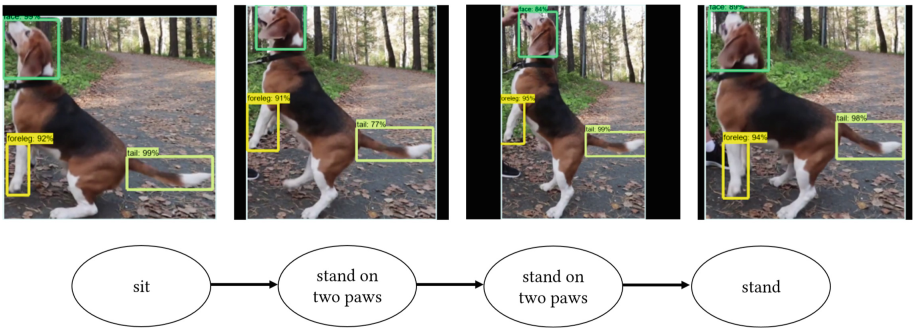
When the End Node is Not Connected to Behavior Nodes. As shown in Fig. 6(c), the end node Stand of the travel route(sit→stand on two paws→stand)



(a) Sample Video 1



(b) Sample Video 2



(c) Sample Video 3

Fig. 6. Example of Behavior Graph travel route created from Input Video

created in sample video 3 is not connected to any behavior node. In order to analyze the behavior of video 3, the similarity between the travel route and posture sequence of all the behavior nodes is measured, and the final behavior is determined as Joyful with the highest similarity as shown in Table 4.

5 Conclusion

In this paper, we proposed a noble framework for help non-expertise to understand dog behavior. Our framework analyzes dog behavior by exploring a graph model that defines successive postures' interrelationships with only a camera without additional high-quality equipment. This process eliminates the need to use multiple sensors and overcomes the limitation of analyzing dog behavior by depending on only a single posture. Our research can be applied to applications such as dog health status and abnormal behavior detection in the future.

References

1. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recogn. Lett. **19**(3–4), 255–259 (1998)
2. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
3. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Anal. Appl. **13**(1), 113–129 (2010)
4. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
5. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
6. Rugaas, T.: On Talking Terms with Dogs: Calming Signals. Dogwise publishing, Wenatchee (2005)
7. Talukdar, J., Gupta, S., Rajpura, P., Hegde, R.S.: Transfer learning for object detection using state-of-the-art deep neural networks. In: 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 78–83. IEEE (2018)
8. Tzutalin, D.: Labelimg. git code (2015)
9. Wu, N., Rathod, V.: Tensorflow detection model zoo (2017)