Motion to Emotion: Animal Emotion Prediction via Pose Estimation

Suin Hwang
Dept. of Computer Science and Engineering
Korea University
Seoul, Korea
suin00h@korea.ac.kr

Seokryun Choi
Dept. of Mechanical Engineering
Korea University
Seoul, Korea
srchoi99@gmail.com

Kwangeun Cho
Dept. of Statistics
Korea University
Seoul, Korea
chopol98@korea.ac.kr

Geunho Kang
Dept. of Data Science
Korea University
Seoul, Korea
rootsquare@korea.ac.kr

***Abstract*— While there are many applications of pose estimation models for humans, there are not many applications for animals. We try to predict animal emotions that are not detected by humans by associating them with motions. We tried to predict the emotions of animals in sequential data with a 2-stage model of emotion predictor that applys the ideas of pose estimation model and transformer. Although the results were not as good as expected, we were able to achieve results with low computing cost by using only keypoint coordinates for the emotion predictor. If we can improve the performance in the future, we can get better results. Data analysis, preprocessing procedure and training of each model are available at** https://github.com/suin00h/motion-to-emotion

Keywords—pose estimation, emotion prediction, r-cnn, transformer

I. Introduction

Inferring the emotion from the action is one of the emerging task in the field of deep learning. Sentiment analysis is one of the examples in natural language processing. Those tasks are straightforward, because it was based on the human's emotion. However, things become hard for animals, since they do not have a verbal/visual language system between them. Moreover, it's hard to track their motions, since they usually have small body than humans. Animal pose estimation was far less likely to be researched than humans. Therefore, analyzing animal pose is a challenging task.

However, these tasks are still meaningful for several reasons. Veterinarians can detect animal's anomaly via emotions. Moreover, as demand of pet grows with the number of single families, this task can provide solutions about reacting with animal's actions. For example, if we put a real time video of the animal into the model, we can easily handle animals.

Therefore, we propose a model which gets emotion from sequential data of animal's action. This model is based on the dataset "Animal Videos for Classifying Pets"[1]. We derives some keypoint locations from the pose estimaiton model, and then use those as input of the emotion prediction model and predicts both action and emotion information. Pose estimation task was performed by keypoint r-cnn which was proposed from Mask R-CNN[2] paper. Also, our data has a sequential feature so the emotion prediction model uses the idea of Transformer[3].

II. Related Work

A. Pose Estimation

Pose Estimation is a task that estimates the position of a keypoint based on image data. Based on the labeled keypoint coordinates, we aim to learn the location of the keypoint in the image.

The method of Pose Estimation is divided into the approachs of top-down and bottom-up. The top-down approach is a method of performing object detection and then estimating the keypoint in it. The task is performed using the backbone network from the object detection part. The botton-up approach estimates the keypoints first and directly connects them to perform the pose estimation task. The keypoints are estimated without using object detection part. In data with a need to estimate many objects' pose, the bottom-up approach, which directly estimates keypoints, is more efficient. However, we adopt the top-down approach at the model, expecting that there is no difficulty in the object detection task due to the data characteristics of one object per image.

Top-down approach pose estimation applys the methods used in object detection and instance segmentation. Object detection typically uses a Featured Pyramid Network (FPN) to extract a feature map of the image, and a Region Proposal Network (RPN) to extract candidate regions of the object[4][5]. In general object detection model, it only performs regression on boxes and classification on objects for detecting the object. But in Pose Estimation, our need is to estimate multiple keypoints, so we use the idea of instance segmentation.

Instance Segmentation based on R-CNN aims to estimate the detected objects on a pixel-by-pixel basis by utilizing the FPN and RPN of the previously created backbone network.[2] The Region of Interest (RoI) generated from the RPN is extracted by RoIAlign from the Feature Map generated from the FPN. RoiAlign performs interpolation on the neighboring cells of the RoI and then averages the results to produce an output. This output is passed to the Fully Connected Network(FCN) behind it, which estimates the pixel-by-pixel mask of the object.

B. *Keypoint R-CNN*

Keypoint R-CNN replaces the whole pixel-by-pixel mask used in Mask R-CNN with a keypoint.[2] In a typical segmentation model, the GT class mask is used to distinguish between object and background, and then each is separated into an object mask and a background mask to predict them.

In Keypoint RCNN, these masks are replaced with the coordinates of each keypoint. The heads of the masks are the coordinates of the keypoints we want to estimate. A bounding box and a classifier are also replaced to predict the keypoints. The detection box becomes the box offsets to find the area of the keypoint, and the classifier determines whether it is a keypoint or not.

In this manner, we need to modify the structure of the model. The training target is one-hot encoded binary mask, which is applied as a single pixel in the most prominent position. *k* keypoints are treated independently, and the goal of training is to reduce the cross entropy loss of the GT keypoint. This produces an output that looks like the shape of the binary mask.

C. *Transformer*

In NLP, recurrent neural networks, long short term memory and gated recurrent neural networks had been a state of the art but commonly couldn't be parallelized because of the sequential nature. To overcome this problem Vaswani et al.[3] proposed new architecture called Transformer with which attention mechanism was an integral part. After the proposal of Transformer, many new architectures have been announced such as BERT, GPT, and ELMo and used in other applications.

D. *ViT*

Inspired by successes in NLP with Transformers, there have been attempts applying Transformer directly to images, which is ViT[6].

First split image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer. Image patches are treated the same way as tokens in an NLP application and used for image classification.

Unlike standard transformer[2] which receives 1D sequence of token embeddings as input, ViT reshape 2D images $x \in R^{H\times W\times C}$ into a sequence of flattened 2D patches $x_p \in R^{N\times(P^2\times C)}$ where P is the resolution of each image patch and $N = H \times W/P^2$ is the resulting number of patches. As in the Transformer, ViT flatten the patches and map to D dimensions with a trainable linear projection. Additionally, similar to BERT's class token, ViT prepends a learnable embedding to the sequence of embedded patches, whose state at the output of the Transformer encoder serves as the image representation y.

III. MODEL

A. *Data*

We only used the data of single dog. Each data video file has roughly hundred frames. For labels of each video file, It had a metadata of overall video. It contains the information of the 15 keypoints of the dog's joint and the location of the bounding box, for each video frame.

For the record, the dataset has a quite unbalanced class distribution in both action and emotion. This will be restated in the prediction part.

B. *Pose Estimation Model*

For each dog image in the video, we calculate and use the dog's 15 joint key-points data in motion/emotion prediction. Each image was expressed in $h \times w \times 3$ tensor, and we get 15×3 tensors from the image.

We used Keypoint R-CNN for this part and stack each joint's prediction into one pile, as we mentioned above.

C. *Emotion Prediction Model*

With the normalized x, y coordinates of keypoints from the pose estimation part, we predict action and emotion using a model inspired by ViT. We flatten a sequence of keypoint coordinates $x \in R^{B\times N\times 15\times 3}$ into $x \in R^{B\times N\times 45}$ and embed it into 256 dim. After concatenating action token and emotion token to embeddings, we add them a positional embeddings

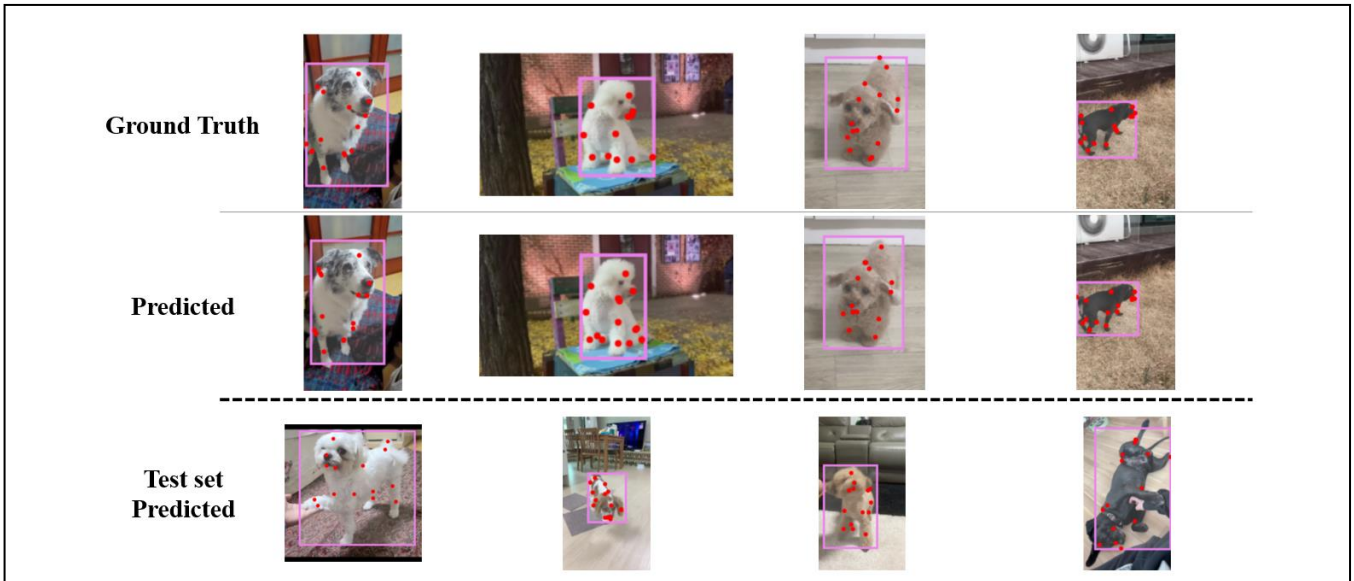

Fig. 1. Results of the Pose Estimation model(Keypoint R-CNN). Above two shows each bounding boxes and keypoints of the ground truth data and predicted output. Below shows another predicted output of some samples from test dataset.

No.	Batch Size	Epochs	Learning Rate	Linear Proj	Val-sample	Class Weight	Pad Mask
1	256	1	2e-5	X	Whole	X	X
2		100		X			
3		1		256			
4		100		256			
5		100		256	20		
6		100	2e-4	512		O	
7		100		1024			O

Table 1. Experiments of training Pose Estimation model.

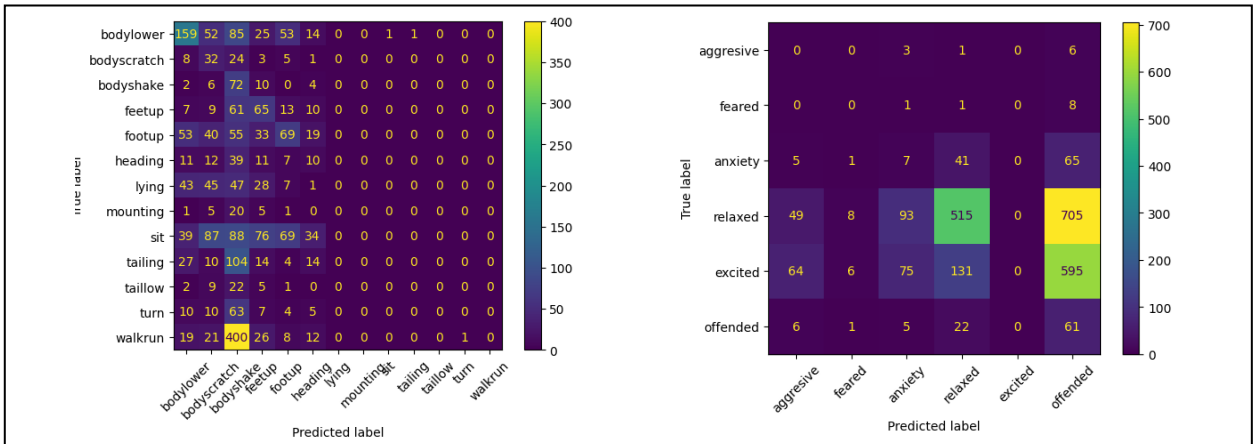

Fig. 2. Confusion matrix of emotion prediction model no.7. Left shows the action class and the right shows

and put the resulting embeddings into a transformer encoder. From the resulting output, we take the first and second tokens and apply shallow MLP layer. After applying softmax function, the results are taken as predicted action and emotion.

IV. Experiment Results

A. Pose Estimation Model

Due to machine performance issues, such as the memory of graphic card resources, we were unable to run the training for a long period of time. However, as shown in the paper in which the model was presented, the accuracy was quite high despite the small number of training, 5 epochs. Figure 1 shows the training results. At this level, it can be said that it is capable of extracting sufficient motion information from images.

B. Emotion Prediction Model

Table 1 shows the seven experiments' conditions during the training. We've done a heuristic search for finding the optimal emotion prediction model. More detailed description of each condition are as follows:

Linear Proj Linear projection layer before the keypoints is projected from 45 dimensions to latent dimensions. During experiments, 256, 512 and 1024 of latent dimensions are used.

Val-sample Before applying the image data, we must deal with different length of the video data. Each datum from the validation set has different video length, but the training dataset has a relatively constant size of length around 20. Therefore, first 20 positional embeddings are trained during rest of the embeddings are not. To overcome this problem, at validation process we sampled at most 20 image frames from the validation set.

Class Weight We found that the dataset has very imbalanced class distribution in both action and emotion, so we measured this by using weighted cross entropy loss.

Pad Mask Since the inputs are varying from their frame length, we needed to make the input sizes uniform by padding. The paddings have no information about the task so we put the pad mask to the transformer encoder. Encoder will ignore the padded part during the inference process.

Figure 2 shows the confusion matrices of the experiment result. The average F1 score for both part is 0.10, which is very poor results. In action prediction, the model completely failed to predict the half of the action types. Most of prediction was misclassified as 'bodyshake'.

V. Conclusion

We have proposed a transformer-based video processing architecture for the dog's emotion prediction task. It looked like it could achieve some decent performance given the successful results of the ViT from which our model has been motivated. But unlike what we might expect, the actual performance was very disappointing, emotion prediction model resulting in poor outputs. Although we couldn't get enough accuracy with a few experiments, there is still room to improve performance by experimenting with more

conditions like removing keypoints normalization, oversampling, source masking, etc.

Our project was also investigating the trade-off between manual feature selection and information loss. The results of our experiments showed that the effect of reduced computational burden using only keypoint information was much lower than the effect of information loss of the images' discarded features. In order to achieve better results on this task in future research, it may be important to design a model that fully utilizes the features of the image.

REFERENCES

[1] https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=59

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. Mask R-CNN. ICCV 2017

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin Attention is All you Need NIPS 2017

[4] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie Feature Pyramid Network for Object Detection CVPR 2017

[5] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun Towards Real-Time Object Detection with Region Proposal Networks NIPS 2015

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby An Image is Worth 16x16 Words: Transformer for Image Recognition at Scale ICLR 2021