Animal Movement Prediction based on the Corroboration of Object Detection and Pose Estimation

Ke Chen [1, †], Yang Yu [2, †, *] and Haoxian Zhou [3, †]

[1] Nanjing University of Information Science and Technology, Nanjing, China

[2] Monash University, Melbourne, Australia

[3] Shantou University, Shantou, China

* Corresponding author email: yyuu0045@student.Monash.edu

[†]These authors contributed equally

Abstract. With the development of automated driving, driving safety has become a hot topic in the field of automated driving. However, the existing methods mainly define living objects as obstacle handling and mostly focus on the safety of pedestrians, thus ignoring the safety of animal movement in automatic driving. In this paper, we propose an animal movement prediction model with the corroboration of object detection and pose estimation, which aims, to help the autopilot perception system judge animal pose more accurately. First of all, we preprocess the datasets, use a Convolutional Neural Network based object detection method to detect animal images, and then use the current mainstream 2D animal pose estimation method to obtain the key points of animal images, and divide the resulting data set into three categories: running, walking and waiting. After preprocessing, we use Resnet and MLP models to classify the pose. For Resnet model, we will evaluate the accuracy of the model on the test set during training and compare the accuracy of the last model to determine the model with the highest training accuracy. The final model has achieved an accuracy of up to 75% for running evaluation. For MLP model, we tend to use the rank of 13 specific bone points' angles as the standard of pose classification. Three types of test sets will be put into three trained MLPRegressor model for training to get the classification we want. Finally, the accuracy of all three poses is more than 80%. Finally, we choose the model with the highest accuracy as our classification method. This work is expected to apply animal pose estimation and animal and pose classification to automated driving to improve driving safety.

Keywords: Animal Movement Prediction; Automated Driving; Pose Estimation.

1. Introduction

With the continuous development of computer vision and deep learning, automated driving has become a hot topic in the development of science and technology, and has received extensive attention from society. In the field of auto driving, one of its cores is the environment perception system, which is a critical enabler for automated driving systems [1]. Environment perception aims to detect obstacles such as vehicles, pedestrians, etc., to help vehicles to avoid them reasonably. Among them, deep learning is widely used in target detection tasks in perception system [2].

At present, vehicle driving safety is a key problem in the field of automated driving, and it is also one of the important factors limiting its development. However, most of the current automatic driving perception systems focus on simple target detection tasks. When encountering pedestrians and animals, they only treat them as obstacles, judge and avoid them according to their positions with vehicles, and ignore the randomness of their movements. Therefore, the introduction of attitude estimation task to target detection can ensure the safety of pedestrians and animals to a greater extent.

Pose estimation in the computer vision field is the specific task of localizing the joint positions or predefined keypoints of an object (e.g. a car, a human body, or a dog) in an image. It goes back to the early works of the 1990s, which were aimed at human detection, motion tracking, and facial pattern estimation [3-4]. For the safety of pedestrians in automated driving, Hariyono et al. [5] propose a video based Vulnerable Road Users (VRU) detection method, which is used to detect vulnerable road users, such as pedestrians and cyclists. An estimated articulated pose can serve as an effective intermediate representation for inferring the direction of motion, intention or gesture of a VRU. Some researchers have proposed a scalable VRU Pose SSD detection and 2-D pose estimation model to optimize the deployment on the automated driving platform [6]. However, they only focus on pedestrian pose detection and estimation on urban roads. The research on human pose estimation is relatively mature, and it has high accuracy in complex scenes, while the research on animal pose estimation is still in its infancy [7]. Considering the safety problems caused by animals crossing the road when the car is driving on the country road, we hope to detect and estimate the pose of animals and establish a pose classification model on this basis to classify the current animal attitudes.

In this paper, we aim to estimate the pose of a large number of animal pictures such as AP-10K dataset [8] by using mmpose to obtain the required training set. The training set is divided into three labels: run, walk and stop, and three corresponding classification models are established. In this way, the vehicle can make early warning by judging the attitude of animals and classifying them into corresponding categories.

2. Method

To correctly identify the animals' action, first we need is to recognize the accurate position of our target animals in the pictures. Since there is no current well-trained identification models of four-legged mammals that fitted our needs, we trained our own model with the help of MMdetection [9]. Then the training images will be put into a MMpose [10] model to get the animals' bone point frame. After all the pre-processing works are done, these pictures will be put into two different routes, one of which tries to use the Resnet network to analyze their raw pixel features and the other one founded a MLP [11-12]network structure to get the answer from the angles of specific bone points. The whole framework of our proposed method is shown in Figure 1, which will be introduced in detail in the following subsections.

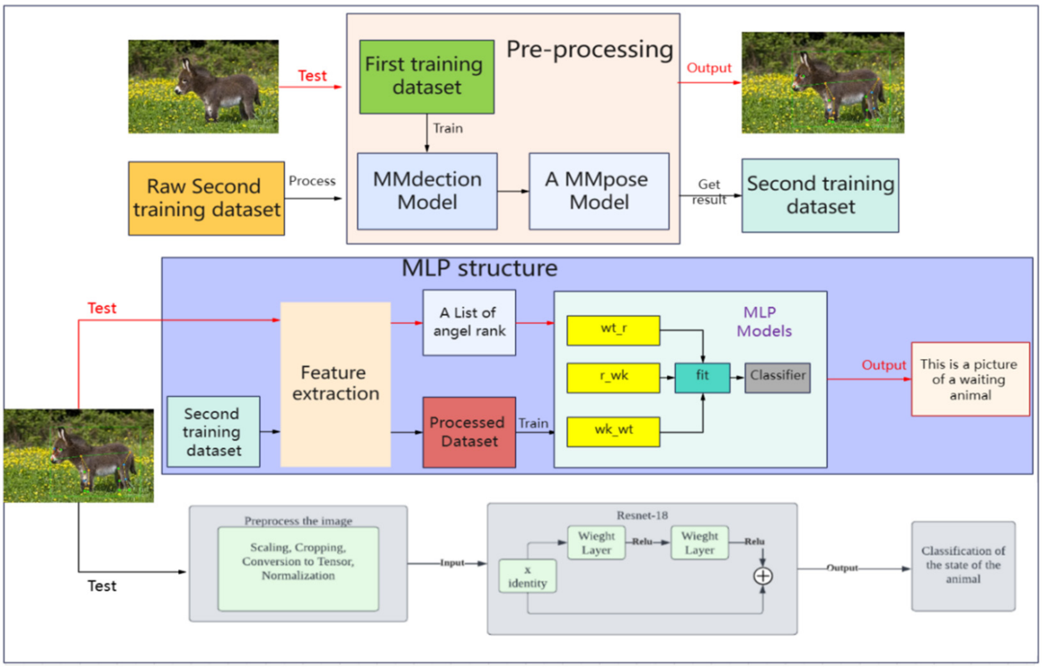

Figure 1. The framework of proposed method

2.1 Image Pre-processing

For the images captured by the camera, we first perform a series of preprocessing including target detection and prediction of key points. We will use the framework covered in OpenMMLab, a complete system of open-source computer vision algorithms developed by Sensetime, to perform this series of image preprocessing. The whole pre-processing consists of an object detection branch and a pose estimation branch. For the former task, we adopt the MMDetection, which is a Pytorch-based deep learning target detection toolbox in OpenMMLab. We will evaluate various models based on our database to find the most suitable model framework in terms of accuracy, efficiency, and volume, and then train them to obtain a model that can perform target detection quickly with guaranteed accuracy. We will find the most suitable model framework in terms of accuracy, efficiency, and volume. After obtaining the target detection model, we will bring the model into another toolbox in OpenMMLab, MMPose, which already has official pre-trained model support for the AP-10K database, so we only need to evaluate the model to obtain the model for key point detection.

2.2 Movement Prediction based on Resnet

After getting the detected images and key point data, on the one hand we will use Resnet directly for its image classification process. Before the modeling training, the images are preprocessed, and forward prediction is performed. This includes scaling and cropping of the training set images, image enhancement (e.g., horizontal flipping), conversion to Tensor, and normalization. Similarly, the images of the test set are scaled, converted to Tensor and normalized.

The dataset is divided into multiple batches before training. During the training of the model, a learning rate reduction strategy (reducing the learning rate at certain Epoch intervals) is inserted. Model building is performed for each batch when training is performed, and the accuracy of the model on the test set is also evaluated. In which a cross-entropy loss function is used to determine if the model is accurate and if there is refinement, as:

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i - [y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)] \quad (1)$$

In the equation, for N data points where y_i is the truth value taking a value 0 or 1 and p_i is the SoftMax probability for the ith data point.

If the new model has a new improvement in accuracy on the test set when comparing the previous model, the previous model is deleted and the new model is recorded. Finally, the one model with the highest accuracy in training is obtained. For model evaluation on the test set, in addition to the accuracy of the model, the model is additionally evaluated with reference to the confidence level of each variable, Re-call value, and f-1score.

2.3 Movement Prediction based on MLP

2.3.1 Feature Selection and Extraction

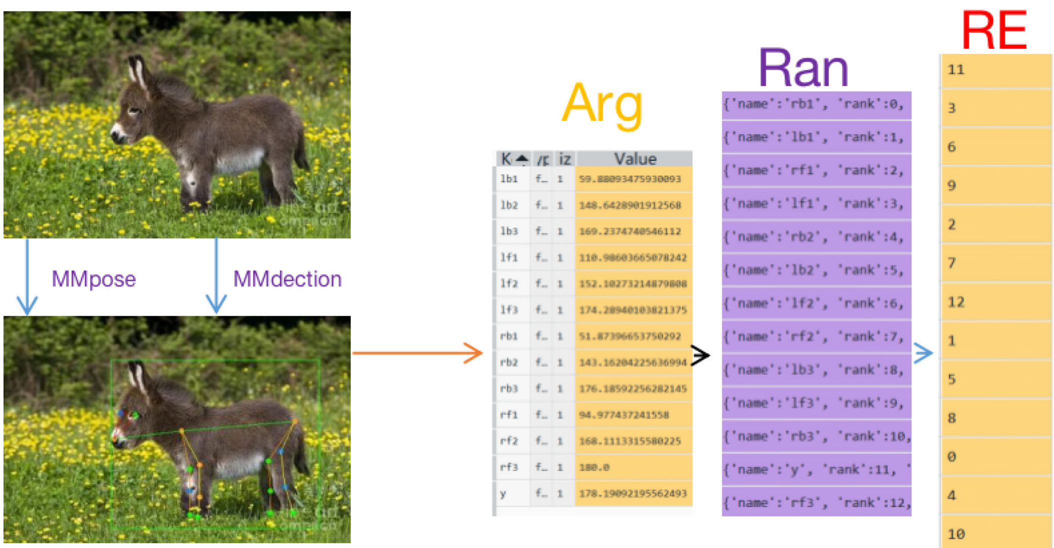

Figure 2. An example of feature extraction

As the Figure 2 shown, based on the detected bounding boxes and estimated key points, we also mine their correlations to design feature representation. Specifically, using the coordinates information of bone points extracted by MMpose, we first successfully got the rank of 13 specific bone points' angles, with 3 for each legs and a special one on animals' necks. The reason why we eventually chose angels' rank as final symbol is for all the mammals, their moving status undoubtedly own a strong relationship with their legs' bending degrees and the simple angular size of a 3-D items' projection on a photo will be affected by the relative position of the photographer to the target.

2.3.2 Model Training

The structure of movement prediction model based on MLP is shown in Figure 3. The three new datasets will be labeled in the form of [wt_r, r_wk, wk_wt], which reflects their similarity degrees to three different actions. For example, all pictures in the wait datasets will own the labels of [0,0.5,1] while the running ones will be [1,0,0.5]. Then, they will be used to train three corresponding MLP Regressor models which are called wt_r model, r_wk model and wk_wt model.

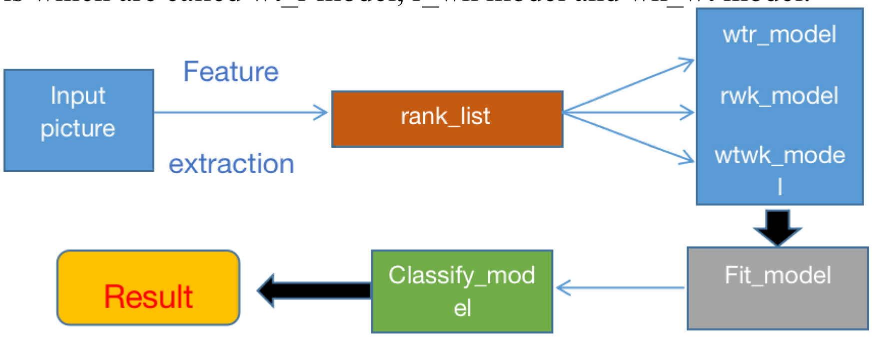

Figure 3. The structure of movement prediction model based on MLP

Raw data in three datasets will be put into three trained MLP Regressor models to get three new labels. Then, three new labels will be used to train a new Regressor model as the input data, while the data that they should be will be put as the symbols. This new model which is called as fit model means to fix the first output data to make them more closerly to their action statuses, which can be seen in Figure 4.

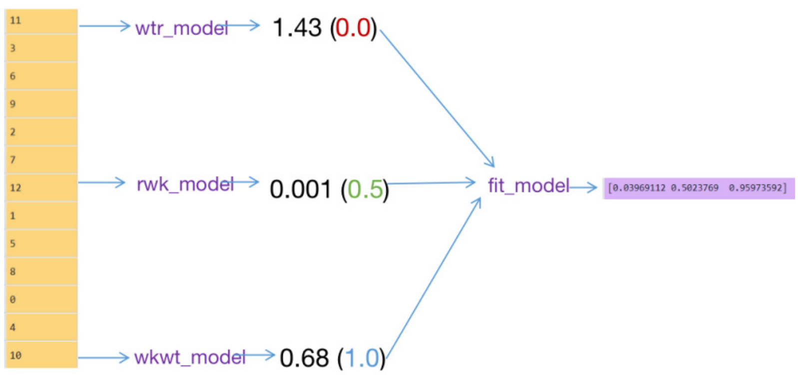

Figure 4. The pipline of fit model

Square difference function is used for all the models above as the loss function, as:

$$J(\theta)=\frac{1}{2}(y-\bar{y})^2 \tag{2}$$

Finally, A MLP Classifier model that could judge the moving status according to the output of the fit model will be trained to realize the final goal.

3. Experiment

3.1 Dataset

To meet the satisfaction of our own research, we collected the 5416 single four-legs animal's pictures from Internet as our Training dateset, which owns 1841 walking ones, 1730 waiting ones and 1845 running ones. Then, 322 walking pictures together with 383 waiting pictures and 368 running pictures composed the Testing datasets. For all the pictures above, only when they could be correctly identified by the two models above that we would use it for coming steps.

3.2 Experiment Setting

To identify the target animals in the pictures, we first used the AP-10k datasets to train a Fast R-CNN [13] model of 50 layers with the help of MMdection. Though limited by the time, we only train the model with a batch size of 6, while its accuracy is much higher than we had believed. 91% of the walking pictures were processed successfully while 85% for waiting ones and 76% for running ones. Then, for bone point frame extraction, we fortunately found the AP-10k group has trained an accurate one which could be directly used for our further experiment. For the MLP structure, we eventually chose sigmoid function as the activation function and Quasi-Newton Methods as the optimizer. For the hidden layers, fit model owns two while three for wtr,rwk and wkwt models.

3.3 Performance Analysis

3.3.1 Results of Resnet 18

We first preprocessed the images, including scaling, enhancement, conversion to Tensor and normalization of the training set. The same RCTN processing (scaling cropping to Tensor normalization) was performed on the test set. When choosing the migration learning strategy depends on the differences between our dataset distribution and the image-net dataset, and since the differences are obvious, we chose to fine tune all training layers. Our dataset is all animals, and the distribution is not quite the same. After that, the average cross-entropy loss function value is calculated for each batch. The obtained values are used to remove the gradient, back propagate and optimize the update.

Figure 5. Visualization of training processing

After building the model, initial tests on the test set yielded an accuracy of about 78.5%. Obviously, this is not satisfactory. So, we made some improvements on this modelling, hoping to make the results of the experiment more accurate. In addition to the cross-entropy loss function, test set accuracy, precision, recall and f1-score are additionally calculated as tools for later evaluation of model accuracy. Some refinements were also made in the training strategy, where the model is trained on each batch. The models are evaluated on the test set directly after training. If the trained model yields a new high accuracy on the test set, the previous model (if it exists) is deleted and the new model is recorded. We also transfer the computed data to WANDB and observe their changes, which is visualized in Figure 5.

The final accuracy rate was refined somewhat to 87%. In the subsequent experiments, we also selected an additional data set for testing, but the results were not particularly accurate during the testing process. The results of different settings are reported in Table 1. In particular, the discrimination between walking and stopping was not particularly accurate. The accuracy for running, walking and stopping was only 0.750, 0.521 and 0.540.

Table 1. Results for different movements based on ResNet-18

	Precision	Recall	F1-score	Support	Accuracy
Run	0.750	0.368	0.494	358	0.368
Wait	0.521	0.537	0.529	383	0.537
Walk	0.540	0.790	0.642	368	0.790
Macro Avg	0.604	0.565	0.555	1109	0.565
Weighted	0.601	0.567	0.555	1109	0.567

3.3.2 Results of MLP

We also conduct a set of experiments to analyze the results of MLP, which can be in Table 2. From the Table 2, we could easily find the three models own a strong ability to recognize the running mammals with 91% accuracy for wtr model and 97% accuracy for rwk model. While when the actions come to waiting or walking, some of the models could not identify them well as only 53% accuracy when wkwt model recognizes the waiting target and 54% for rwk model recognizing the walking mammals. Obviously, a running animal own a strong feature in bones' angel rank and that's why running action could be easily identified among the three actions. What's more, we find all the three models own a high accuracy to identify one of their target action while a much lower accuracy for the other one. This phenomenon truly caught our interest for it undoubtedly own a strong relationship with the internal algorithm of the models which deserves for further research.

Table 2. Results for different movements based on three MLP models

	wtr	rwk	wkwt
run	0.907	0.977	/
wait	0.703	/	0.531
walk	/	0.538	0.957

In the last prediction result of the Classifier model, we find the accuracy reaches a satisfactory degree just as what Table 3 shows

Table 3. Results for the whole structure

Total accuracy	0.841
The accuracy of run	0.854
The accuracy of wait	0.841
The accuracy of walk	0.831

4. Discussion

From the experiment above, the MLP structure owns a higher accuracy when it comes to identifying mammal's actions, which may reveal animals' moving status has a strong relationship with their bone points' angles that are line with our daily cognition. While as what we could find in both of the methods that a running mammal is much easier to be found than both of the others. This phenomenon truly caught our interest for when we tagged the pictures in our datasets, we also found it was hard to classify a picture into a waiting one or a walking one. After preliminary discussions, we realized it is unrealistic to judge a mammals' waiting or walking action just rely on a single picture, which means a moving video could be much better.

For further research, more types of actions would be created such as jumping or sleeping. Since we didn't own enough time to create a big enough dataset, we plan to collect more pictures with more animal's types. Though the system now could only identify the pictures with one mammal inside, we hope to improve it to own the ability to work with plural targets. To our satisfaction, the final result has met our initial requirement which is meant to solve a potential problem in the field of autonomous driving. What's more, we hope our work could be used in more areas like farm management or animal monitoring in protected parks.

5. Conclusion

In this paper, we propose two classification models based on animal pose estimation. We divide these into three steps according to the final goal, including using a target detection method based on Convolutional Neural Network to detect animal images, using mainstream 2D animal pose estimation methods to obtain key points of animal images, and using Resnet and MLP network to construct classification models of three main animal poses. The two models classify the pose according to different standards. We also discuss some major difficulties in improving the accuracy of classification models, and explored some possible solutions. A series of large and repeated experimental results show that our model has high accuracy in judging the classification of three animal attitudes, and basically meets the requirements for users to input animal pictures to judge and classify their attitudes. This method, which combines the animal pose estimation and classification model, converts the less obvious animal attitude estimation composed of key points and lines into the text for prompt output of the corresponding pose category, which is more intuitive and easy to judge. We expect that this method will be applied to the automated driving of cars to help the automated driving perception system judge the animal pose more accurately. Although we are optimistic about the development prospect of this method, there is still a lot of room for further development.

References

[1] C. Guindel, D. Martín and J. M. Armingol, "Joint object detection and viewpoint estimation using CNN features," 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2017, pp. 145-150, doi: 10.1109/ICVES.2017.7991916.

[2] Pathak A R, Pandey M, Rautaray S. Application of deep learning for object detection[J]. Procedia computer science, 2018, 132: 1706-1717.

[3] Yang G., Huang T.s. Human face detection in a complex background Pattern Recognit, 27(1994), pp. 53-63.

[4] Sung K.-K., Poggio T. Example-based learning for view-based human face detectionIEEE Trans. Pattern Anal. Mach. Intell., 20 (1998), pp. 39-51.

[5] Hariyono, J.; and Jo, K.-H. 2015. Pedestrian action recognition using motion type classification. In Proc. IEEE CYBCONF, 129–132.

[6] Kumar C, Ramesh J, Chakraborty B, et al. Vru pose-ssd: Multiperson pose estimation for automated driving[C]//Proceedings ofthe AAAI Conference on Artificial Intelligence. 2021, 35(17): 15331-15338.

[7] Jiang Le,Lee Caleb, Teotia Divyang, Ostadabbas Sarah. Animal pose estimation: A closer look at the state-of-the-art, existing gaps and opportunities[J]. Computer Vision and Image Understanding,2022,222.

[8] Yu H, Xu Y, Zhang J, et al. Ap-10k: A benchmark for animal pose estimation in the wild[J]. arXiv preprint arXiv:2108.12617, 2021.

[9] Mahajan, A., & Chaudhary, S. (2019, June). Categorical image classification based on representational deep network (RESNET). In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 327-330). IEEE.

[10] Pal, K. K., & Sudeep, K. S. (2016, May). Preprocessing for image classification by convolutional neural networks. In 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 1778-1781). IEEE.

[11] Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.

[12] Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors). MIT Press, 1986.

[13] Ross Girshick; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.