강아지 행동 분석을 위한 YOLOv4 기반의 실시간 객체 탐지 및 트리밍

오스만*, 이종욱**, 박대희**†, 정용화**
*고려대학교 컴퓨터정보학과 **고려대학교 컴퓨터융합소프트웨어학과
e-mail: osumanatif@gmail.com

YOLOv4-based real-time object detection and trimming for dogs' activity analysis

Othmane Atif*, Jonguk Lee**, Daihee Park**†, Yongwha Chung**
*Dept. of Computer Information Science, Korea University
**Dept. of Computer Convergence Software, Korea University

Abstract

In a previous work we have done, we presented a monitoring system to automatically detect some dogs' behaviors from videos. However, the input video data used by that system was pre-trimmed to ensure it contained a dog only. In a real-life situation, the monitoring system would continuously receive video data, including frames that are empty and ones that contain people. In this paper, we propose a YOLOv4-based system for automatic object detection and trimming of dog videos. Sequences of frames trimmed from the video data received from the camera are analyzed to detect dogs and people frame by frame using a YOLOv4 model, and then records of the occurrences of dogs and people are generated. The records of each sequence are then analyzed through a rule-based decision tree to classify the sequence, forward it if it contains a dog only or ignore it otherwise. The results of the experiments on long untrimmed videos show that our proposed method manages an excellent detection performance reaching 0.97 in average of precision, recall and f-1 score at a detection rate of approximately 30 fps, guaranteeing with that real-time processing.

1. INTRODUCTION

The recent breakthrough in computer vision and deep leaning has led to a large interest in human and animal activity monitoring systems. Following that trend, in a previous work of ours, we presented a system that automatically monitors the activities of dogs left alone at home and informs owners in case behaviors that are deemed potentially harmful to the dogs or their environments are detected [1]. For testing purposes, pre-trimmed video data containing a dog alone was used as input for the system. However, in a real-life situation, such system would be running continuously and receive untrimmed video data, including instances that do not contain a dog, ones that contain people only, and ones that contain the dog and people. In that case, feeding the system with such data would be a waste of computational resources and is most likely to affect its detection accuracy. Thus, it is important to make sure that the data received contains a dog with no people around it. In this study, we propose a system to automatically trim the continuous video data received into short sequences of frames and use a deep learning detector followed by a decision tree to select the ones that include a dog and forward them to the activity recognition system in real-time.

Many studies worked on detecting objects for the purpose of monitoring humans and animals. In some of them, motion detection was employed to ensure that the frames are not empty. For example, in their work, Fang et al. [2] aimed to detect animals in aerial videos using the optical flow to detect motion, and velocity threshold to distinguish the foreground pixels from the dynamic background. While it can be helpful in some cases, using motion detection to detect animals is a vulnerable method that can lead to false detections, especially in a setting like ours where dogs can be motionless at times and where the motion could be resulting from humans. Other work recognized the effectiveness of deep learning-based detectors to perform object detection. One example is the work presented by Buric et al. [3] where they compared the performance of two detectors, namely Mask R-CNN and YOLOv2, to apply as a pre-requisite for action detection in videos involving handball players through a frame by frame detection. This is similar to what we are trying to achieve, but instead of using Mask R-CNN like they did, we believe newer detectors would perform better. In fact, more recent works are in favor of newer releases of YOLO. For instance, Ju et al. [4] used YOLOv3 to detect turkeys in an enclosed area to identify and monitor them individually as a first step to analyze their behaviors and interactions, and in their paper, Lu et al. [5] used an improved version of YOLO to perform real-time detection

† Corresponding author

of vehicles in videos.

In this paper, we make use of the latest release of YOLO, YOLOv4, which guarantees an optimal accuracy and speed of detection [6], to perform detection of dogs and people in videos. However, unlike previous works where detectors processed videos frame by frame and used the detection result of each frame individually, we first trim sequences of frames from the video, perform a frame by frame object detection on each sequence and use a rule-based decision tree to classify it based on the detection results of all its frames. By doing that, we ensure that false negative frames inside a sequence are not dropped, since the sequence is classified as a whole, which helps conserve useful motion information needed by the monitoring system when analyzing the dog's activities. The system proceeds by trimming and then feeding the frames of each sequence to the YOLOv4 model, and the detections are used to decide on the class that the sequence belongs to through a rule-based decision tree.

2. PROPOSED METHOD

The overall architecture of the system we are proposing is shown in Figure 1. The system includes three modules: *the data collection module*, *the trimming module*, and *the decision-making module*.

2.1 Data Collection Module

The data collection module is where the video data received in real time from the camera gets preprocessed in 2 stages, namely rate sampling and frame cropping. After that, the preprocessed frames are forwarded to the next module.

2.2 Object Detection and Trimming Module

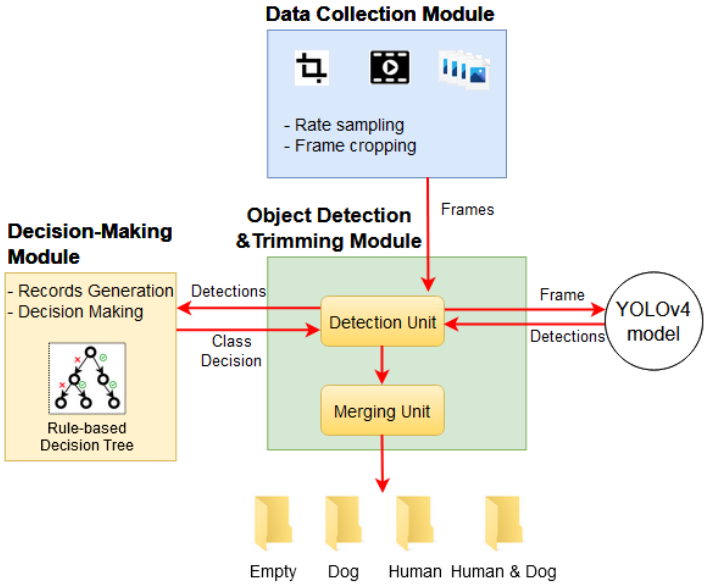
The frames received from the data collection module are trimmed in sequences, then fed one frame at a time to the YOLOv4 model to perform object detection. YOLO has been a popular and effective detector since it was first introduced by Redmon et al. [7] for its low latency, making it ideal for real-time detection. After a series of upgrades, Bochkovskiy et al. [6] released its 4th version, an improvement that has proven to be faster and more accurate than other existing detectors.

For each sequence of frames, the detection unit prepares a set of corresponding detections by performing a frame by frame object detection using the trained YOLOv4 model. When an object detection result is returned from the YOLO model, the detection unit rearranges it into a list of detections, since some frames contain multiple detections, to allow easier processing. The set of detections for the sequence of frames is then forwarded to the decision-making module where the class to which the sequence belongs is selected. After receiving the class decision from that module, the detection unit would forward the sequences containing a dog only to the activity recognition system described in [1]. However, in order to evaluate our trimming system as a stand-alone one, we added a merging unit. This unit receives both the frame sequence and the decision, combines each sequence into a video file and saves it in the folder whose name matches the class decision.

2.3 Decision-Making Module

As stated previously, after the detection unit receives the detections for a sequence, they are sent to the decision-making module where they are analyzed through 2 consecutive phases. The first phase aims at extracting the occurrence of each class in every frame of the sequence in order to generate what we refer to as records. Records are basically counters used to track information about the detections in each sequence of frames, such as how many frames are empty and in how many frames each object is detected. For each frame, the module checks the corresponding detection to verify if it is empty or contains detections of 1 or 2 of the objects. If no class is detected, the record for empty frames is incremented. If a frame has a single detection, the relevant class' record, i.e. human or dog, is incremented and in the case of multiple detections in a single frame, detections belonging to the same class result in only one incrementation of the class record. At the end of the first phase, the generated records would depict for each sequence how many empty frames there are and in how many frames each of humans and dogs are present at least once.

In the second phase, the decision-making module feeds the records of the sequence to a decision tree that returns the class decision of that sequence of frames. This class decision is then forwarded to the detection unit of the trimming module, and then to the merging unit as described in the previous paragraph.



(Figure 1) General architecture of the YOLOv4-based real-time trimming system for dogs' videos.

3. EXPERIMENTAL RESULTS

3.1 Data Collection Experiments

We originally gathered video data to prepare a dataset for dogs' activity recognition and the process is described in more details in [1]. Since then, more videos were recorded following the same settings and safety measures with a different dog. We currently have multiple videos containing one of the two dogs alone and occasionally with people walking around in a safe space with instances where one or more would walk out of the camera range. Dogs in these videos have been spatially annotated using ViTBAT [8].

In order to build a dataset for object detection using YOLOv4, we first used ViTBAT to label people in some of the recorded videos since now we are interested in detecting both dogs and people. After that, we used our script to extract

images of both dogs and people from the video files and convert the annotation format to the YOLO one. Empty frames were also added to the dataset with empty labels to serve as negative data. To guarantee the diversity of the data, frames were taken from different videos where both people and dogs had different outfits. For the empty data, it was collected using diverse settings of the room by moving some furniture around. Figure 2 below shows examples of frames used in the dataset.

(a) Empty

(b) Person

(c) Dog

(Figure 2) Examples of images used to prepare the dataset for object detection using YOLOv4.

Table 1 below shows the details of the dataset we used to train our YOLOv4 for object detection. The dog data is composed of frames containing one of the two dogs, while the person one includes frames of 4 people, 2 men and 2 women.

<Table 1> Image dataset used for YOLOv4 object detection model training

Class	Total (frames)
Dog	20735
Person	17344
Empty	25622
Total	63701

3.2 Object Detection using YOLOv4

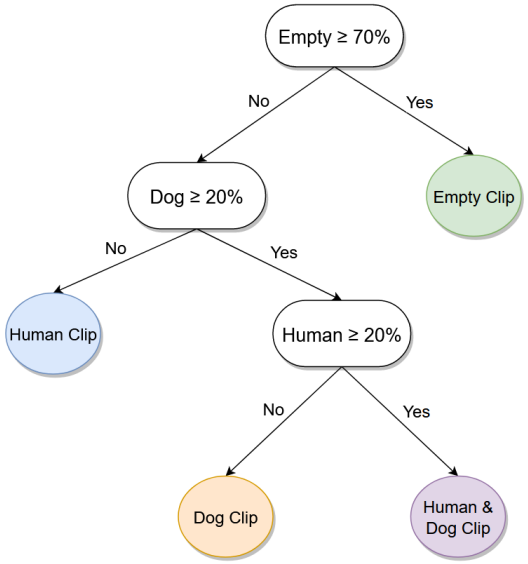
The YOLOv4 object detector was trained using the darknet framework implementation provided by Bochkovskiy et al. [6] as it is periodically improved and contains all the tools we needed. After building the framework in our environment, we used the official YOLO labelling tool to confirm that the labels were correctly converted. The reasons why we used our script to convert the ViTBAT labels instead of directly using the YOLO tool are because, first, many videos were already labelled using ViTBAT, and second, the YOLO tools requires a frame by frame labelling and does not offer the option of labelling videos. Thus, using the YOLO tool would have been time consuming and would have resulted in redundant work.

After the detection model training was done, we compiled the dynamic link library (DLL) of darknet to enable loading and using the model in our code. The system was then implemented, and the trained model was deployed in it.

3.3 Trimmed Videos Classification using Rule-based Decision Tree

After the data collection module performs sampling on the video data with a rate of 27 fps, it crops some unwanted area from each frame and sends them to the next module. Then, the video data received by the trimming module is trimmed in sequences of 27 frames, equaling a video clip of 1 second. This ensures both an accurate trimming and fast processing, and results in the decision-making module receiving a set of 27 lists of detections that are used to generate occurrence records. As we are targeting 2 objects, the module generates 3 records: empty, person, and dog. Although some scenarios include multiple occurrences of an object class in the same frame (e.g. 2 or 3 people in the room), we are only interested in the occurrence and not the specific number of objects. Thus, if more than 1 human or more than 1 dog are detected in the same frame, the corresponding record is only incremented once.

The records are then fed to the decision tree shown in figure 3 where thresholds of records are used to decide on the class each sequence of 27 frames belongs to.



(Figure 3) Rule-based decision tree for classification of sequences of frames based on their objects' occurrence records.

The records of empty frames are used as the root node because they are the only ones that constitute a Boolean attribute. The frame can either be empty or not empty (dog and/or human(s)). The threshold was set to 70%, meaning that if at least 19 out of 27 frames have no detection, the sequence is classified as empty. If the sequence is not classified as empty, the percentages of both person records and dog records (from the total of the non-empty frames) are analyzed according to the following rules:

$$Dog \geq 20\% \ \& \ Human \geq 20\% \rightarrow Human \ \& \ Dog \quad (1)$$

$$Dog \geq 20\% \ \& \ Human \leq 20\% \rightarrow Dog \quad (2)$$

$$Dog \leq 20\% \ \& \ Human \geq 20\% \ \rightarrow Human \quad (3)$$

The thresholds can be adjusted to guarantee a stricter or a more flexible decision. In our case, we decided to make use of a 20% threshold in order to compensate for possible false and no object detection cases.

3.4 Implementation Details

The darknet framework was built with Visual Studio 2015 and OpenCV 3.4.0 in a computer using Windows 10, a CPU of type Intel i5-8500, 32 GB of RAM and a graphic card GTX 1080Ti. The YOLOv4 model was trained using a leaning rate of 0.001, 32 subdivisions, 64 as batch, 416×416×3 as input size and the max_batch was set as 60000. A model was saved after every 1000 iteration to be kept as backup. The training and testing data were divided using an 8:2 ratio on the object detection image dataset.

The same environment was used to implement and test the whole system with the YOLOv4 model saved at the 52000^{th} iteration as the loss was stable after the 48000^{th}. Python 3.6.4 was used to implement the system and the scripts we used.

3.5 Video Trimming Results

In order to evaluate the system, we used 5 untrimmed videos of lengths varying between 2 and 30 minutes with scenarios alternating between empty room, either one dog or human(s) only, and both one dog and human(s). The videos were from different recordings than the ones from which the dataset to train the YOLO model was made, and they were manually labeled to specify the temporal boundaries of the objects. We also implemented an evaluator module that uses the labels of the untrimmed videos as ground truth to evaluate the results of the proposed system. Table 2 below summarizes the trimming results using different metrics, namely the precision, recall and f1-score. The classification performance of all 4 classes exceeded 0.90 on all the metrics used. The average precision, recall and f-1 score reached 0.97 and the speed of detection using the trained YOLOv4 model was almost 30fps, making it suitable for real-time detection. This confirms that our proposed system achieves very accurate and fast results when trimming videos based on the presence of dogs and humans. Therefore, it can be used as a preprocessing module to automatically select sequences that contain dog only from incoming video data in real-time in order to forward it to the activity detection system in [1].

<Table 2> Trimming results of the proposed method

Class	Precision	Recall	F-1 score	Support
Empty	1.00	0.99	1.00	1220
Dog	1.00	0.99	0.99	1628
Human	0.91	0.98	0.94	216
Human & Dog	0.97	0.95	0.96	309
Accuracy	-	-	**0.99**	3373
Macro Average	**0.97**	**0.98**	**0.97**	3373
Weighted Average	**0.99**	**0.99**	**0.99**	3373

4. CONCLUSION

In this paper, we proposed and implemented a system that automatically trims long videos into short sequences, detects the presence of dogs and people and select the sequences of frames classified as containing a dog only. We trained a YOLOv4 detection model on images of dogs and people, and then designed a rule-based decision tree to classify sequences of frames based on the results of frame per frame detections. We also provided experimental results to show the accuracy of our proposed method in trimming long videos into short clips and classifying them in real-time based on the presence or absence of dogs and people in the frames.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A3B07044938 and NRF-2020R1I1A3070835).

References

[1] O. Atif, J. Lee, D. Park, and Y. Chung, “Camera-based Dog Unwanted Behavior Detection System,” Korea Information Processing Society (KIPS), Seoul, 2019.

[2] Y. Fang, S. Dub, R. Abdoolaa, K. Djouania, and C. Richardsa, “Motion Based Animal Detection in Aerial Videos,” 2nd International Conference on Intelligent Computing, Communication & Convergence, Procedia Computer Science 92, p 13-17, Bhubaneswar, 2016.

[3] M. Burić, M. Pobar, and M. Ivašić-Kos, “Object detection in sports videos,” 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1034-1039, Opatija, 2018.

[4] S. Ju, M.A. Erasmus, A.R. Reibman, and F. Zhu, “Video Tracking to Monitor Turkey Welfare,” 2020 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), pp. 50-53, Albuquerque, 2020.

[5] S. Lu, B. Wang, H. Wang, L. Chen, L. Ma, and X. Zhang, “A real-time object detection algorithm for video,” Computers & Electrical Engineering, Vol. 77, pp 398–408, 2019.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhad, “You only look once: Unified, real-time object detection,” arXiv preprint arXiv 1506.02640, 2015.

[8] T. A. Biresaw, T. Nawaz, J. Ferryman, and A. Dell, “ViTBAT: Video Tracking and Behavior Annotation Tool,” IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 295–301, 2016.