WEP Warwick Evans Publishing

Transactions on Engineering and Technology Research
ISSN: 2960-1797, eISSN: 2960-2246 | Volume 1
AMMEE 2023

Real-time Pedestrian Tracking Based on YOLOv3 and Prototype Clustering

Ruopeng Li

School of Mathematical Sciences, Beihang University, Beijing 100191, China

Abstract. Although the accuracy of existing neural network models is high, in pedestrian tracking tasks, due to the uncertainty of targets, when tracking new targets, it is necessary to fine-tune the model, which further requires large computing and storage resource overhead. Therefore, its application on some lightweight platforms, such as robots and UAVs, is limited. Pedestrian tracking by robots and UAVs still faces great challenges in occlusion, multi-target, target loss, etc. This paper mainly solves the problem of real-time pedestrian tracking by object detection model of robot lightweight, which is mainly based on YOLO network to detect pedestrians, and then proposes a novel lightweight model and prototype clustering algorithm. Numerous experiments on the ETH dataset validate the superiority and effectiveness of our approach.

Keywords: Real-time Pedestrian Tracking, YOLO Network, Lightweight model.

1. Introduction

The goal of the Person Re-ID task is to identify a particular pedestrian's image from a large set of images. At present, the vast majority of methods to complete Person Re-ID tasks are based on deep learning. Existing Person Re-ID models designed based on deep learning focus on feature extraction [1], and often combine global features with local features.

The traditional Person Re-ID model needs to transmit the target image into the neural network fine-tuning discriminator after a certain frame is selected, and then apply it to the next frame to locate the target; at this time, the located target will be used for fine-tuning the discriminator again, and so on until the re-recognition task is completed. In this way, the model needs to be trained when processing each frame of the video stream, which requires continuous computing power consumption. Lightweight platforms equipped with low-performance processors, such as robots and drones, may not meet the computing power requirements. This paper proposes a lightweight model and prototype learning algorithm to reduce the computation involved in processing video streams.

We use the pre-trained model under yolov3 to extract all pedestrians from the video stream as samples to be classified, and output decision boxes and global features. Next, we design a lightweight network based on prototype clustering to find the target's decision box from these decision boxes.

2. Related Work

The proposed Person Re-ID method can be classified into 4 classification methods. According to the type of recognized object, it can be divided into image-based Person Re-ID [2] and video-based Person Re-ID [3].; According to the method type, it can be roughly divided into deep learning methods [4] and metric learning methods [5].; According to whether there is supervision, it can be divided into supervised method and unsupervised method[6]; According to the application scenarios, it can be roughly divided into conventional Person Re-ID [7], occlusion scenario Person Re-ID [8], cross-resolution Person Re-ID [9], domain adaptive person re-id [10], cross-modal person re-id [11] and clothes changing Person Re-ID [12].

This paper mainly solves the problem of real-time pedestrian tracking in moving scenes. Complex models based on deep learning cannot meet the real-time requirements, while traditional measurement methods have low accuracy. Therefore, Person Re-ID based on moving scenes often faces a dilemma of "accuracy-real-time". This paper aims at the dilemma of "precision-real-time" and achieves a high-performance balance through a lightweight model.

3. Methodology

3.1 Pedestrian Detection and Feature Alignment

The baseline for pedestrian detection in this paper is YOLO [13], using the YOLOv3 version. The YOLO model is suitable for pedestrian detection and feature extraction in moving scenes due to its fast inference speed and relatively high accuracy. Let the YOLO model be M and the input sample be x. After inference of the model M, the pedestrian output box set B and the corresponding feature box set F are obtained, i.e.

$$B,F=M(x) \tag{1}$$

Let the number of pedestrians detected in this input image x be m, then a set of m pedestrian boxes is B= [B1, B2, . . ., Bm], where anyone pedestrian box Bi={(xl, yl), ((xr, yr))} (i=1, 2, . . . , m) in B is represented by coordinates representing upper left and lower right corners of the box. Similarly, the feature set corresponding to each set of pedestrian boxes is $F = [F_1, F_2, \ldots, F_m]$, wherein each feature representation is cut from a position of the pedestrian box corresponding to the feature map.

After each pedestrian frame in the image is obtained, m pedestrian samples can be cut out from corresponding positions in the image x according to the position of the frame and denoted as $P = [P_1, P_2, \ldots, P_m]$. Similarly, features are also cut out from feature maps at corresponding positions. However, since the height and width of each pedestrian in the image are different, the intercepted pedestrians and their features have different sizes. To facilitate input into the subsequent lightweight network, we need to scale these pedestrian frames and features to a fixed size, that is, normalize them. In this paper, we use the method of ROIAlign[] to normalize and align pedestrian boxes and features. The normalized pedestrian samples and features are denoted as P', F' respectively.

3.2 Lightweight Model

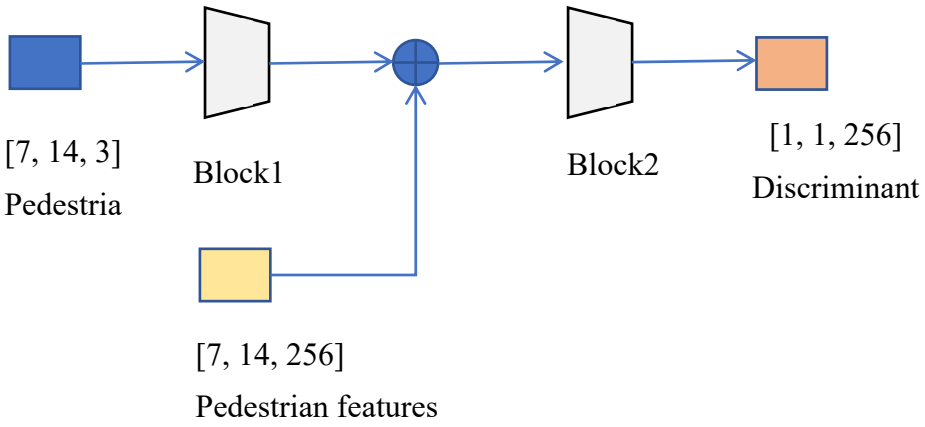

Figure 1. Construction of Lightweight Model

We first designed a lightweight model based on pedestrian image input and its feature input, as shown in Fig. 1. First, the normalized pedestrian image is input into the first block of the model for feature extraction, and then this feature and pedestrian features are further fused and input into the second block. After feature pooling, the final discriminant line features are obtained. In the model, the first block consists of three convolutional layers: 1X1, 3X3 and 1X1. Each convolutional layer is followed by a Batch Norm layer and a Relu layer, with characteristic channels of 512,256 and 256 respectively. In the feature fusion operation, element-by-element addition is adopted. The second block is like the first block, except that the feature channels are 256, 256 and 256 respectively. After outputting the features, AvgPolling pooling operation is used to change the features into 1X1X256 size features.

The light weighting model is only trained in the initial phase, using a VIPeR dataset. During training, a classifier is connected behind the lightweight model for pedestrian recognition. The data set is first

input into YOLOv3 to obtain features and pedestrian boxes, which are then input into the lightweight model for model training. After the training, the classifier is no longer used. Only YOLOv3 and lightweight model are used for Person Re-ID task.

3.3 Prototype Clustering

In this section, we construct a target pedestrian prototype using the Gaussian Mixture Model (GMM) approach. For the pedestrian target tracking task, we collect N images containing targets as a Gaussian mixture model tracker. First, the N images are input into YOLOv3 model to obtain corresponding N pedestrian samples P' and feature map set F'. Then the pedestrian samples and feature maps are input into the lightweight model to obtain N final features, which are denoted as $f = [f_1, f_2, \ldots, f_N]$. Next, the target prototype is constructed.

3.3.1 GMM Model Initialization

Let the number of prototypes be K. First, initialize K Gaussian models to form a Gaussian Mixture Model (GMM), and randomly initialize K Gaussian component mean values [μ1, μ2, ..., μN], variance [σ1, σ2, ..., σN] and weight parameters [α1, α2, ..., αN]. In this paper, we use the EM algorithm to build a Gaussian mixture model of the target,

$$p(x) = \sum_{k=1}^{K} \alpha_k \mathbb{N}(x|\mu_k \sigma_k) \quad (2)$$

3.3.2 Feature Matching

For the new pedestrian feature f_{new}, it is input into the Gaussian mixture model for prediction to obtain the prediction probability p corresponding to the sample feature, and a threshold value is set. When the prediction probability is greater than the set threshold value, we consider the pedestrian corresponding to the sample feature as the target pedestrian.

3.3.3 Model Update

We consider the target pedestrian features predicted by the model and use them to update the current GMM model, which can make the GMM model more robust. If we directly add the current features to the previous target feature set to recalculate the new GMM, it will greatly increase the time of the whole detection and tracking process, which is contrary to our goal of real-time pedestrian tracking. In this paper, we use a sliding update mechanism to update the statistics of Gaussian model in the following form:

$$\mu_k = \beta\mu_k + (1-\beta)f \quad (3)$$

$$\sigma_k = \gamma\sigma_k + （1-\gamma）|f-\mu_k|$$

4. Experiment

4.1 Experiment Setup

4.1.1 Data Set

In this experiment, the ETH dataset is used to validate the hybrid Gaussian-based feature matching algorithm proposed in this paper. ETH is a data set for pedestrian detection. The test set contained 1804 images from three video clips. The dataset was captured from a car-mounted stereo device with 640 x 480 (bayered) resolution and frame rate of 13-14 FPS.

Unlike other data sets that collect images from multiple cameras, ETHZ collects images from moving cameras. Although the viewpoint variation is relatively small, it does have considerable illumination variations, scale changes and occlusion. With reference to the method of Learning Discriminative Appearance-Based Models Using Partial Least Squares, a total of 146 target pedestrians are intercepted from three videos in the dataset ETH, and each target pedestrian is stored in one target

video sequence. The video sequence contains an average of 60 picture frames and other pedestrian picture, with a total of 100 frames.

4.1.2 Model Setting and Experimental Environment

In this experiment, the YOLOv3 model [YOLOv3] trained based on CoCo dataset [14][CoCo] is used for pedestrian detection, and the YOLOv3 model is loaded with a python-based OpenCV module. This method can be used for fast model inference without GPU to achieve real-time pedestrian detection and feature extraction.

To facilitate the subsequent feature matching, we perform ROIAlign [MaskRCNN] alignment on the extracted features corresponding to pedestrian boxes. Finally, all features corresponding to pedestrian detection frames are resized to 32X64X256 resolution size, and then flattened into a one-dimensional vector for subsequent tracker construction. In the process of constructing a tracker, each target pedestrian extracts 20 images containing the target from its target video sequence as the tracker construction. The number K of Gaussian models in the Gaussian mixture model is set to 4. In the feature-to-Gaussian mixture model matching, we set pedestrians corresponding to features with a probability greater than 50% as targets.

4.1.3 Metrics

For this experimental metric, the F-score obtained by "Precision (P)-Recall (R)" is used. Take a pedestrian target as an example,

(1) The number of target pictures contained and predicted as targets is denoted by TP;

(2) The number of pictures containing the target that are not predicted as targets is denoted FN;

(3) The number of targets that do not contain target pictures and are predicted as targets is denoted by FP;

(4) The number of targets that do not include target pictures and are not predicted as targets is recorded as TN.

Then, the accuracy of this target is

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Recall rate of this target

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

The F-score for this target is

$$\mathrm{F} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}$$

Finally, the F values corresponding to 146 all target pedestrians are averaged to obtain an average F-score, i.e. mF-score.

4.2 Test Results and Analysis

4.2.1 Routine Test

On the ETHZ dataset, for each of the 146 target video sequences, 20 frames with targets are extracted to update the Gaussian Mixture Model, and then YOLOv3 model is used to detect pedestrians in each frame of the video, extract pedestrian boxes and corresponding feature vectors, input the feature vectors into the Gaussian Mixture Model for prediction, and finally integrate all video sequence results. The results are shown in Table 1. A total of ten experiments were carried out to obtain the

mean and variance of the final indicators. Our method achieves an average detection and tracking rate of 31 frames per second while ensuring high accuracy.

Table 1 Real-time Pedestrian Tracking Results on ETHZ Dataset

Accuracy P (%)	Recall Rate R (%)	mF-score(%)	Efficiency (FPS, frames per second)
0.9042±0.0347	0.8729±0.0816	0.8883±0.0582	29.57±3.05

4.2.2 Ablation Experiment

Effect of k. We validate the effect of the number of Gaussian models in a Gaussian mixture model on experimental results. We set k = 1 (single Gaussian model), 2, 3, 4, 5. The experimental results are shown in Table 2. Setting the number of Gaussian models in different Gaussian mixture models has a significant impact on the results. In this experiment, when k is set to 4, the best effect is achieved. Although the accuracy of single Gaussian model is also very high, it is significantly lower than the mF-score value of 4 Gaussian model, and the result does not become significantly better when k is greater than 4. Therefore, we choose k=4 as the benchmark in routine experiments.

Table 2 Effect of k on Experimental Results

K	1	2	3	4	5
mF-score(%)	0.8179	0.8527	0.8593	0.8883	0.8874

Impact of sampling strategy. We validate the effect of the sampling strategy for each target sample on the results when initializing a Gaussian mixture model. We consider two sampling strategies: (1) randomly sampling a fixed number of target samples to update the Gaussian mixture model; (2) manually sampling targets with multiple perspectives and differentiated fixed sample numbers. Among them, we consider 20 samples for each target. Random sampling is to extract 20 frames of image samples from all samples containing targets by random non-replay sampling method, while manual sampling considers different angles, distances, illumination changes and other factors of the target to make the samples more diversified as much as possible. The experimental results are shown in Table 3. It is obvious that the manual sampling method has a higher accuracy. Manual sampling can make the model more robust in the case of small sample sizes, but it greatly increases costs if large sample sizes are considered. Therefore, if the sample size is large, random sampling method should be adopted and a NMS-like sampling algorithm should be designed to ensure that the new samples can maximize the current sampling difference as much as possible.

Table 3 Impact of Sampling Strategy

Sampling Strategy	Random Sampling Method	Manual Sampling Method
mF-score(%)	0.8307±0.1422	0.8883±0.0582

Impact of sampling quantity. In this experiment, we verified the influence of different sampling sizes on the experimental results under different sampling strategies. Since the average number of video frames containing targets in each sample video sequence is about 60, we set the maximum sampling quantity to be 45 and the sample rate to be about 75% for initialization of Gaussian mixture model. Because the sampling rate is too high and there are too few test samples left, it will have a great impact on the results and affect the determination of experimental conclusions. The number k of Gaussian mixture models is set to 4. The experimental results are shown in Table 4. The experimental results are less affected by manual sampling, while random sampling depends on the number of samples. The larger the number of samples, the higher the accuracy of the random sampling-based method. This is because many samples will make the sampled target image more likely to cover all possible poses of the target, thus resulting in the learned Gaussian mixture model being more representative. Manual sampling can ensure the diversity of samples when there are fewer samples, and the Gaussian mixture model has higher performance.

Table 4 Effect of Sampling Quantity

Sampling Quantity	1	10	20	30	45
Random Sampling mF-score (%)	0.6021±0.8849	0.8072±0.7861	0.8497±0.5427	0.8504±0.3681	0.8598±0.2422
Manual Sampling mF-score (%)	0.6237±0.3412	0.8382±0.2872	0.8593±0.1422	0.8597±0.0972	0.8802±0.0318

5. Conclusion and Prospect

5.1 Conclusion

In this paper, a real-time pedestrian tracking algorithm based on YOLOv3 is proposed. Through lightweight model construction and prototype clustering algorithm for fast and efficient feature matching, not only high accuracy of pedestrian tracking can be achieved, but also the inference speed can reach real time. The proposed algorithm is fully validated on the ETHZ dataset.

5.2 Prospect

In this paper, the feature extraction of pedestrian frame is mainly based on YOLOv3, and a lighter and more accurate pedestrian detection method is considered. For further feature matching, it is also considered to be integrated into the model without separately constructing a Gaussian mixture model. At the same time, a combination of pedestrian detection and pedestrian tracking can be considered, while considering the coordination of accuracy and speed.

References

[1] An Feng-Ping, Liu Jun-E, Pedestrian Re-identification Algorithm Based on Visual Attention-positive Sample Generation Network Deep Learning Model..Information Fusion. Volume 86-87, Issue . 2022. PP 136-145

[2] Zheng Feilong, Research on Person Re-ID Algorithm Based on Dynamic Images, Information and Computer (Theoretical Edition), 2019(03)

[3] Chen Yang. Research on Video-based Person Re-ID Method. Huazhong University of Science and Technology, 2019

[4] Luo Hao, Jiang Wei, Fan Xing et al. Research Progress of Person Re-ID Based on Deep Learning. Chinese Journal of Automation, 2019, 45(11): 2032-2049.

[5] Huang Haixin, Tao Wenbo, Du Tingting, Overview of Person Re-ID Based on Metric Learning, Journal of Shenyang University of Technology, 2023, 42(05)

[6] Tang Jiamin, Han Hua, Huang Li, Coarse and Fine Grain Feature Extraction Based on Unsupervised Learning in Pedestrian Recognition, Computer Engineering. 2022, 48(04)

[7] Chang Zhe, Research on Pedestrian Re-recognition Algorithm for Complex Scenarios, Central South University, 2022

[8] Zhao Cairong, Qi Ding, Key Technologies of Intelligent Video Surveillance: A Review of Pedestrian Re-recognition Research. Science in China: Information Science, 2021, 51(12).

[9] Geng Yanbing, Lian Yongjian, Cross-resolution Person Re-ID Based on Multi-granularity Feature Generation Adversarial Networks, Computer Applications, 2022, 42(11)

[10] Chen Peixian, Research on Domain Migration Methods for Person Re-ID, Xiamen University, 2021.

[11] Bian Ziyang, Li Jianan, Cross-modal Person Re-ID Method and Application Based on Edge Intelligent Terminals, Chinese Journal of Ordnance Industry, February 26, 2023

[12] Zhang Peixu, Hu Guanyu, Yang Xinyu, A Domain Enhanced and Adaptive Paradigm for Person Re-ID in Dress Change, Journal of Xidian University, August 2023

[13] Zhang Mingzhen. Underground Pedestrian Detection Model Based on Dense-YOLO Network, Industrial and Mining Automation, 2022, 48(03)

[14] Guan Jiacheng. Improved Lightweight Target Detection Based on YOLOv5, Computer System Applications, 2023, 32(09)