Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 2
CII 2023

Robot Real-time Pedestrian Tracking Algorithm based on Deep Learning

Ziheng Qu *

Electrical Engineering and Automation, City Institute, Dalian University of Technology, Dalian, 116600, China

* Corresponding Author

Abstract. In recent years, deep learning models have greatly improved the performance of pedestrian detection and tracking. Although the accuracy of neural network-based modeling methods is high, they often require large computational and storage resource overheads, which makes them difficult to apply to robots with high resource requirements. Pedestrian tracking for robots still faces great challenges in the problems of occlusion, multi-target, and target loss. In this thesis, we focus on solving the problem of real-time pedestrian tracking by lightweight fusion of deep learning target detection models for robots, firstly, through the lightweight YOLO network, we perform pedestrian detection and feature extraction, and then we propose a Gaussian mixture model based feature matching method to construct the target pedestrian tracker, and finally, we use the PID-based control algorithm for real-time control of the robot's motion, and finally realize the real-time pedestrian Tracking. In this thesis, we validate the feature matching method based on the Gaussian mixture model on the ETH dataset, and at the same time, combined with the motion control algorithm, we carry out the actual validation, and the experimental results show that our proposed method can realize the real-time pedestrian tracking.

Keywords: Deep Learning; Real-time Pedestrian Tracking Algorithm; Artificial Intelligence.

1. Introduction

The simple idea that "what you see, you know" led naturally to the development of computer vision. As computer technology continues to improve, computer vision has also developed rapidly. In the past few years, the emergence of deep learning has led to breakthroughs in the field of computer vision, as reflected in image classification [1] target detection [2], and target segmentation [3].

Target detection is an important research problem in the field of computer vision, which has a wide range of applications, including object tracking [4], automatic driving [5], intelligent security [6], and so on. Aiming at the problem of target detection, this paper utilizes Yolov3, a target detection framework with high efficiency, to perform target pedestrian detection and feature extraction for the purpose of pedestrian alignment. Through continuous optimization on the base model, Yolov3 has achieved great improvement in accuracy and speed and can play a more important role in practical application scenarios.

The focus of this research is how to realize real-time pedestrian tracking using a lightweight model and improve the accuracy and speed of tracking by studying pedestrian tracking algorithms and applying them in real-world scenarios to meet people's needs.

First, we use the Yolov3 pre-trained model as the core of pedestrian detection. Yolov3 is an efficient target detection model that is trained based on the Coco dataset. It not only features high accuracy and efficiency but also has a great advantage in lightweight, which can meet the demand of real-time pedestrian tracking. Secondly, we design a pedestrian re-identification matching algorithm based on a hybrid Gaussian model, which is combined with the tracker to realize the real-time pedestrian tracking algorithm. This algorithm can effectively avoid wrong tracking caused by target occlusion, illumination changes, and image noise. At the same time, it can improve the accuracy and robustness of tracking and realize more accurate pedestrian tracking. Due to its lightweight computation, it can

achieve higher speed matching and real-time tracking. Finally, we carry out robot motion control based on the PID motion control algorithm, the main purpose is to make the robot adjust the steering in the left and right directions and the speed in the forward and backward directions in real-time according to the motion estimation of the target by the motion control algorithm, so as to achieve that the target can always appear in the robot's centered field of view.

2. Methodologies

2.1. Overview of the Methodology

In this section, pedestrian detection and feature extraction based on Yolov3, feature matching and tracker based on Gaussian mixture Gaussian model, and motion control methods are respectively introduced.(as shown in Figures 1 and 2)

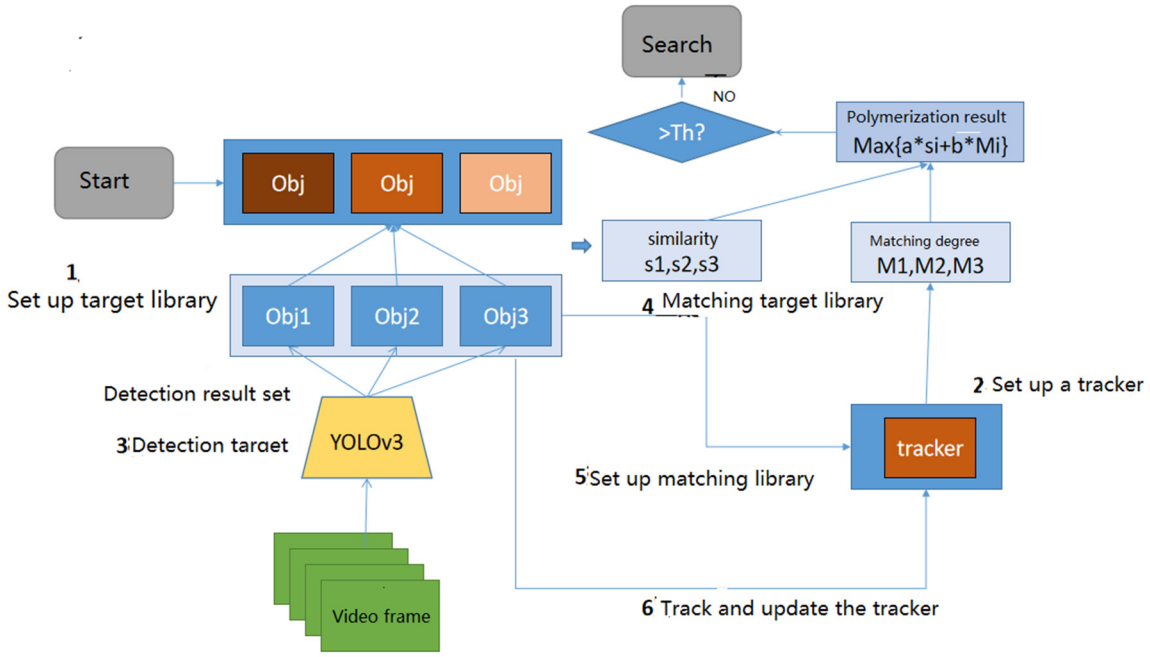

Figure 1. Technical proposal

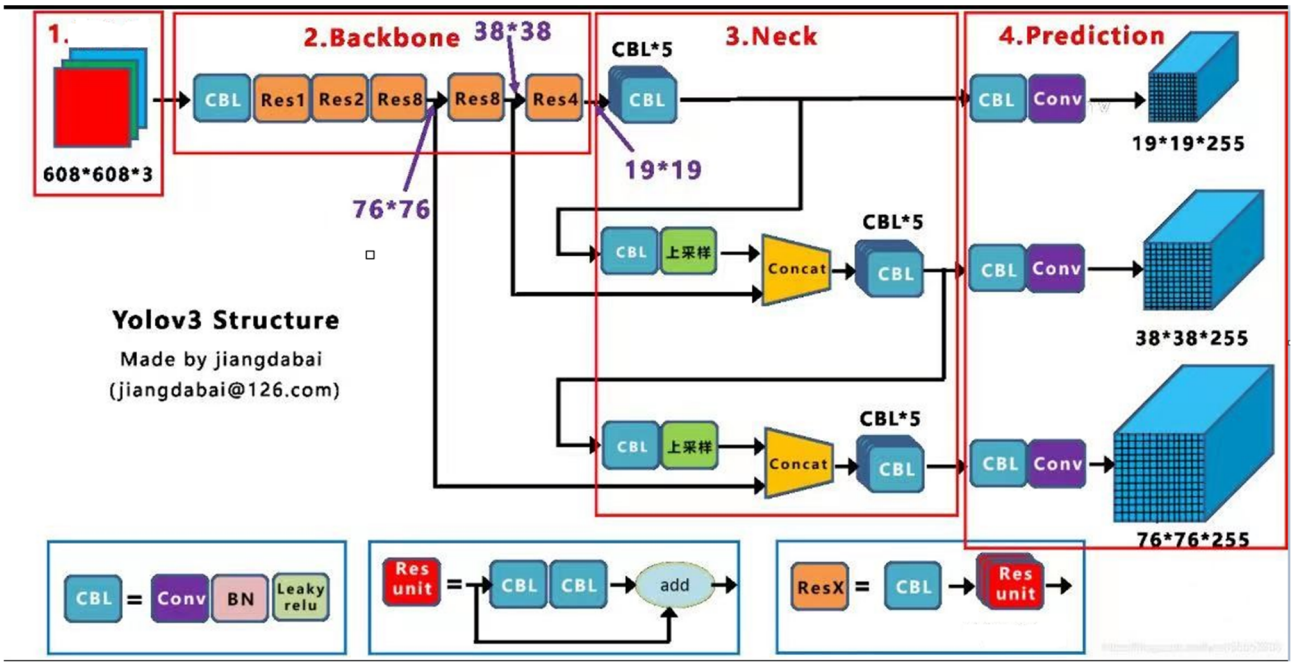

Figure 2. Motion Control Schematic

2.2. Yolov3 Synopsis

Yolov3 (You Only Look Once version 3) is a target detection algorithm that is efficient and fast. It employs a one-shot forward propagation approach to directly predict the locations and categories of multiple targets in a single image, hence the name "you only look once".

The modeling framework of Yolov3 is based on a convolutional neural network (CNN), which consists of a backbone network and auxiliary layers [7]. The backbone network usually employs some pre-trained deep convolutional neural networks, such as Darknet-53, for extracting features from images. The auxiliary layers include a feature pyramid network and a detection layer for target detection at different scales.

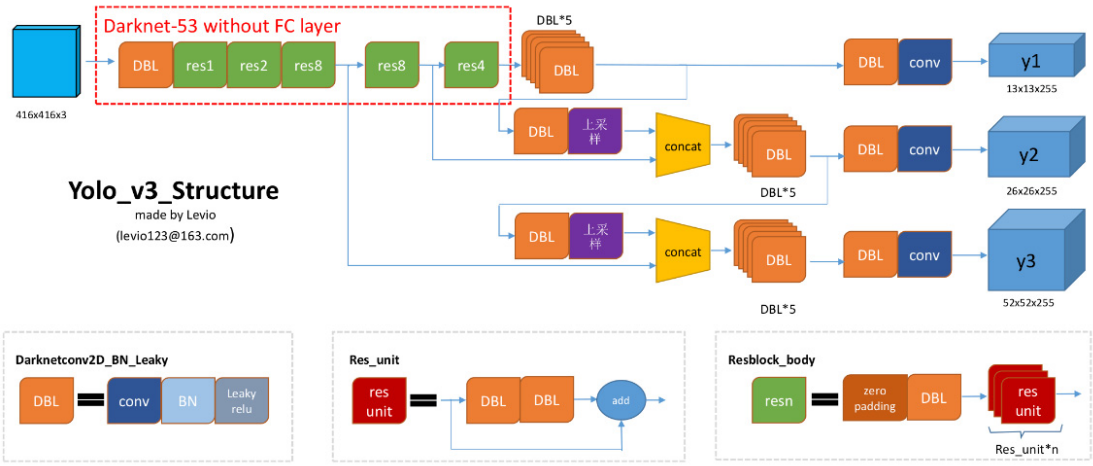

Figure 3. Yolov3 modeling framework

The training methodology of Yolov3 consists of two phases: pre-training and fine-tuning. The pre-training phase uses a large-scale dataset (e.g., ImageNet) to train the features extracted from the backbone network. The fine-tuning phase uses a specific target detection dataset, such as Coco or Voc to fine-tune and optimize the model according to the target detection task.

Yolov3 functions to perform real-time target detection and localization. It can detect the location and class of multiple targets simultaneously and output their bounding boxes and confidence levels. Yolov3 strikes a good balance between detection speed and accuracy and is well-suited for applications in real-time scenarios, such as video surveillance and autonomous driving.

Putting Yolov3 on a robot can be accomplished by loading the model using OpenCV's DNN (Deep Neural Networks) module, which provides the ability to load and run deep learning models, using the interface provided by OpenCV to load the trained Yolov3 model and perform real-time target detection on the robot. The detected targets and their location information can be obtained by capturing images or video streams through the camera, feeding them into the model, and then parsing the model output. In this way, the robot is able to sense and recognize target objects in the surrounding environment in real-time.

For the input video frame I, the expression for pedestrian detection and feature extraction based on Yolov3 is given in the following equation:

$$P, F = YOLOv3$$

where P denotes the set of pedestrian frames detected by the model in the input image I, i.e., $P=\{P_1,P_2,\ldots,P_m\}$, m denotes the number of detected pedestrians, and each pedestrian frame is a rectangle consisting of the coordinates of the upper-left and lower-right corners of the frame, i.e., $P_i=\{(x_l,y_l),(x_r,y_r)\}(i=1,2,\ldots,m)$. F denotes the feature corresponding to each pedestrian frame, which

is combined by three different scale output features of Yolov3. In this paper, we use the ROIAlign method in the MaskRCNN[8] method for feature alignment to Resize the feature resolution to 32X64X256 size to facilitate subsequent feature matching.

2.3. Gaussian Mixture Model Feature Matching based Tracker

For the task of pedestrian target tracking, we collect N images containing the target as the construction of a Gaussian mixture model tracker. First, these N images are input into the Yolov3 model to obtain the corresponding N feature atlases $F= \{F_1,F_2,... ,F_N\}$.

2.3.1. Model Initialization:

We use K Gaussian models to form a Gaussian mixture model (GMM), and randomly initialize the K Gaussian component means $[\mu_1, \mu_2,...,\mu_N]$, variance $[\sigma_1,\sigma_2,... ,\sigma_N]$ and weight parameters $[\alpha_1,\alpha_2,... ,\alpha_N]$. In this paper, we use the EM algorithm for Gaussian mixture model construction and finally get the Gaussian mixture model of the target.[9][10]

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

2.3.2. Feature Matching

For the new pedestrian feature f, input it into the Gaussian mixture model for prediction, get the prediction probability p corresponding to this sample feature, set the threshold threshold, when the prediction probability is greater than the set threshold, we consider the pedestrian corresponding to this sample feature as the target pedestrian.[11][12][13]

2.3.3. Model Update

We consider that for the target pedestrian features predicted by the model, they will be used to update the present 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 the goal of real-time pedestrian tracking that we pursue. In this paper, we use the sliding update mechanism to update the statistics of the Gaussian model in the following form:

$$\mu_k = \beta\mu_k+(1-\beta)f \quad (2)$$

$$\sigma_k = \gamma\sigma_k+(1-\gamma)|f-\mu_k| \quad (3)$$

2.4. Motion Control

The PID controller adjusts the output of the controller according to the current error (the difference between the set value and the actual value) to achieve stable control of the system.

The output of the PID controller consists of three parts:

(1). Proportional: This item adjusts the controller's output proportionally, which is proportional to the current error. It determines the speed and stability of the response, too large a proportional gain may cause overshooting, and too small a gain may lead to a long response time.

(2). Integral: The integral of the accumulated error over a period of time is used to offset the static error of the system and to eliminate steady-state deviations. The integral term is slower to eliminate errors but ensures the accuracy of the system when it reaches the set point.

(3). Derivative: This term predicts the future trend of the error based on the rate of change of the error and is used to suppress system oscillations and improve stability. Derivative terms can reduce the overshoot of the system and reduce the response time, but sensitive to noise.

Combining these three terms, the output of the PID controller can be calculated by the following equation:

$$\text{Output} = K_p * \text{Error} + K_i * \text{Integral (Error)} + K_d * \text{Derivative (Error)} \quad (4)$$

where K_p, K_i and K_d denote the proportional, integral, and differential gains respectively, Error denotes the current error, Integral(Error) denotes the integral value of the error, and Derivative(Error) denotes the derivative value of the error.

We carried out an analog simulation on Matlab, took the real-time video-based target pedestrian tracking results as inputs, adjusted the PID parameters for real-time target pedestrian tracking, and finally carried out the actual test.

3. Experiment

3.1. Experimental Setup

3.1.1. Data Set

This experiment uses the ETH dataset for validating the hybrid Gaussian-based feature matching algorithm proposed in this paper.ETH is a dataset for pedestrian detection. The test set contains 1804 images from three video clips. The dataset was captured from a stereo device installed in a car with a resolution of 640 x 480 (bayered) and a frame rate of 13-14 FPS.

Unlike other datasets that collect images from multiple cameras, ETHZ collects images from moving cameras. Although the point-of-view variations are relatively small, it does have considerable illumination variations, scale variations, and occlusions. We refer to *Learning Discriminative Appearance-Based Models Using Partial Least Squares* for a total of 146 target pedestrians intercepted from the three videos in the dataset ETHZ, each stored in a target video sequence that on average contains 60 image frames of the target and other pedestrian image frames, totaling 100 images.

3.1.2. Model Setup and Experimental Environment

This experiment uses the Yolov3 model [Yolov3] trained on the Coco dataset[14] for pedestrian detection, while the python-based OpenCV module is used to load the Yolov3 model. This method allows for fast model inference in a GPU-less environment for real-time pedestrian detection and feature extraction.

In order to facilitate the subsequent feature matching, we perform ROIAlign[MaskRCNN] alignment operation on the extracted features corresponding to the pedestrian frames, and finally, all the features corresponding to the pedestrian detection frames are Resize to 32X64X256 resolution size and then flatten them to one-dimensional vectors for the subsequent construction of the tracker. During the construction of the tracker, each target pedestrian is constructed as a tracker by extracting 20 images containing the target from its target video sequence. The number of Gaussian models in the Gaussian mixture model, K, is set to 4. In feature-Gaussian mixture model matching, we set the pedestrians corresponding to features with a probability greater than 50% to be the targets.

3.1.3. Metric Criteria

In this experiment, we use the F-score value obtained from "Precision (P)-Recall (R)" as the metric criterion. Take a pedestrian target as an example.

(1) The number of images that contain the target and are predicted to be the target is denoted as TP;

(2) The number that contains the target picture and is not predicted as a target is denoted as FN;

(3) The number that does not contain the target picture and is predicted to be a target is denoted as FP;

(4) The number that does not contain the target picture and is not predicted as the target is denoted as TN.

Then, the precision rate of the target is

$$P = \frac{TP}{TP+FP} \tag{5}$$

The recall of this objective is

$$R = \frac{TP}{TP+FN} \tag{6}$$

The F-score value for this goal is

$$F = \frac{2PR}{P+R} \tag{7}$$

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

3.2. Experimental Results and Analysis

3.2.1. Conventional Experiments

On the ETHZ dataset, for each target sequence of the 146 target video sequences, 20 of the video frames with targets are extracted to update the Gaussian mixture model, then the Yolov3 model is used to detect the pedestrians in each frame of the video, and the pedestrian frames and the corresponding feature vectors are extracted, and then the feature vectors are inputted into the Gaussian mixture model to make the prediction, and finally, the results of all the video sequences are integrated, and the results are shown in Table 1. The results are shown in Table 1. We conducted a total of ten experiments to obtain the mean and variance of the final metrics. 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 the ETHZ dataset

Precision rate P(%)	recall rateR(%)	mF-score(%)	efficiency(FPS,Frames per second)
0.8842±0.0347	0.8529±0.0816	0.8683±0.0582	31.57±2.05

3.2.2. Ablation Experiments

Effect of k. We verified the effect of the number of Gaussian models in the Gaussian mixture model on the 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 effect on the results, and in this experiment, the best results are achieved when k is set to 4. The single Gaussian model, although also highly accurate, is significantly lower than the mF-score value of the 4-Gaussian model, and the results do not become significantly better when k is greater than 4. Therefore, we choose k=4 as the benchmark for our regular experiments.

Table 2. Effect of k on experimental results

K	1	2	3	4	5
mF-score(%)	0.8179	0.8527	0.8593	0.8683	0.8674

Sampling strategy effects. We validate the effect of the sampling strategy for each target sample in initializing the Gaussian mixture model on the results. We consider two sampling strategies: (1) randomly sampling a fixed number of target samples to update the Gaussian mixture model, and (2) manually sampling a fixed number of target samples with multiple perspectives with differentiation. Among them, we consider sampling 20 samples for each target, and random sampling is to take 20 frames of picture samples by random no-return sampling method in all the samples containing the target, while manual sampling is to make the samples more diversified as much as possible by taking into account the different angles of the target, the distance and proximity, and the change of illumination. The experimental results are shown in Table 3. It can be clearly seen that the manual sampling method has a higher accuracy rate. In the case of small sample size, manual sampling can

make the model more robust, but it will greatly increase the cost if a large sample size is considered. Therefore, we recommend using random sampling if the sample size is large and designing an NMS-like sampling algorithm to ensure that new sampling samples maximize the current sampling variability as much as possible.

Table 3. Impact of sampling strategy

sampling strategy	random sampling method	Artificial sampling method
mF-score(%)	0.8307±0.1422	0.8683±0.0582

Effect of sampling quantity. In this experiment, we verified the effect of different sampling quantities under different sampling strategies on the experimental results. Since the average number of video frames containing the target in each sample video sequence is around 60 frames, we set the maximum number of samples to be 45, which is about 75% of the sample sampling rate, for the initialization of the Gaussian mixture model. This is because too high a sampling rate leaves too few test samples, which can have a relatively large impact on the results and affect the determination of the experimental conclusions. The number of Gaussian mixture models k is set to 4. The experimental results are shown in Table 4. It can be seen that the experimental results are less affected by the use of manual sampling, while random sampling is more dependent on the number of samples. The larger the number of samples, the higher the accuracy of the random sampling-based method, this is because a large number of samples will make the sampled target image more likely to cover all possible poses of the target, which will lead to a more representative learned Gaussian mixture model. Manual sampling, on the other hand, ensures a diversity of samples when there are fewer samples on the sample, and the higher the performance of the Gaussian mixture model.

Table 4. Effect of number of samples

sample size (statistics)	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	0.8602±0.0318

4. Summary and Outlook

4.1. Summary

In this paper, a real-time pedestrian tracking algorithm based on Yolov3 is proposed, which can not only realize high accuracy of pedestrian tracking but also achieve real-time inference speed by constructing the tracker through a hybrid Gaussian model and feature matching through cosine similarity. The proposed algorithm is fully validated on the ETHZ dataset.

4.2. Outlook

This paper is mainly based on Yolov3 for feature extraction of pedestrian frames. Future work considers to construct a lightweight feature extraction network to further input the features detected by Yolov3 for pedestrian frames or the corresponding features of pedestrian frames into this lightweight model, so as to improve the discriminative and robustness of the features. Further feature matching is also considered to be incorporated into the model without the need to construct a separate Gaussian mixture model.

References

[1] Choi DongHee, Analyzing angiogenesis on a chip using deep learning-based image processing, Lab on a chip. Volume, Issue. 2023.

[2] Wenqing Yao, Sheng Li, Review of target detection algorithms based on deep learning, Science & Technology Information, 2023,21(16).

[3] Cui Mengyang, Research and Implementation of Target Segmentation Algorithm Based on Deep Learning, Xidian University.

[4] Liu Jiaqi, Semantic SLAM Based on Dynamic Object Tracking, Computer Application Research, 2023，07(05).

[5] Ma Jianhong, Review of Image and Point Cloud Fusion Methods in Autonomous Driving, Journal of Zhengzhou University (Science Edition). 2022,54(06).

[6] Wang Dazhi, Multi-sensor acquisition and data fusion for intelligent monitoring of target detection, communication power technology. 2020,37(12).

[7] LV Chen, Cheng Deqiang, Kou Qiqi, et al. Target tracking algorithm based on yolov3 and ASMS. Opto-Electronic Engineering,2021,48(02):70-80.

[8] Dorrer M G;Alekhina A E. Normalization of data for training and analysis by the MaskRCNN model using the k-means method for a smart refrigerator's computer vision. Journal of Physics: Conference Series. Volume 1889, Issue 2. 2021.

[9] Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP. "Section 16.1. Gaussian Mixture Models and k-Means Clustering". Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press.2007.

[10] Yu, Guoshen. "Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity". IEEE Transactions on Image Processing.2021. 21 (5): 2481–2499.

[11] Wolfram Research. "CosineDistance – Wolfram Language & System Documentation Center". wolfram.com.2007.

[12] COSINE DISTANCE, COSINE SIMILARITY, ANGULAR COSINE DISTANCE, ANGULAR COSINE SIMILARITY. www.itl.nist.gov. Retrieved 2020-07-11.

[13] P.-N. Tan, M. Steinbach & V. Kumar, "Introduction to Data Mining", Addison-Wesley (2005).

[14] LV Chen, Cheng Deqiang, Kou Qiqi, et al. Target tracking algorithm based on yolov3 and ASMS. Opto-Electronic Engineering,2021,48(02):70-80.