# Target Detecting and Target Tracking Based on YOLO and Deep SORT Algorithm

Jialing Zhen[1], Liang Ye[1,2(✉)], and Zhe Li[3]

[1] Department of Information and Communication Engineering, Harbin Institute of Technology, Harbin 150080, China
yeliang@hit.edu.cn
[2] Health and Wellness Measurement Research Group, OPEM Unit, University of Oulu, 90014 Oulu, Finland
[3] China Academy of Launch Vehicle Technology, Beijing 100076, China

**Abstract.** The realization of the 5G/6G network can ensure high-speed data transmission, which makes it possible to realize high-speed data transmission in the monitoring video system. With the technical support of 5G/6G, the peak transmission rate can reach 10G bit/s, which solves the problems of video blur and low transmission rate in the monitoring system, and provides faster and higher resolution monitoring pictures and data, and provides a good condition for surveillance video target tracking based on 5G/6G network. In this context, based on the surveillance video in the 5G/6G network, this paper implements a two-stage processing algorithm to complete the tracking task, which solves the problem of target loss and occlusion. In the first stage, we use the Yolo V5s algorithm to detect the target and transfer the detection data to the Deep SORT algorithm in the second stage as the input of Kalman Filter, Then, the deep convolution network is used to extract the features of the detection frame, and then compared with the previously saved features to determine whether it is the same target. Due to the combination of appearance information, the algorithm can continuously track the occluded objects; The algorithm can achieve the real-time effect on the processing of surveillance video and has practical value in the future 5G/6G video surveillance network.

**Keywords:** Target detecting · Target tracking · Deep convolutional neural network · Kalman filter

## 1 Introduction

At present, in the field of target tracking, tracking algorithms are mainly divided into two categories, Generative Algorithm and Discriminant Algorithm. The generative algorithm is to first establish the target model or extract the target features [1], and then search for similar features in subsequent frames to track. In recent years, the relevant representative algorithms include Kalman Filter [2], Mean Shift, ASMS, and so on [3], Discriminant Algorithm [4] means that the target model and background information are taken into


S. Shi et al. (Eds.): 6GN 2021, LNICST 439, pp. 362–369, 2022.
https://doi.org/10.1007/978-3-031-04245-4_32

account at the same time. The current frame takes the target area as the positive sample and the background area as the negative sample. The machine learning method [5] trains the classifier. In recent years, the related representative algorithms include TLD, SVM, and so on [6].

## 2 Target Detecting Based on YOLO V5s

The YOLO network is mainly composed of three main parts [7]:

- Backbone: a convolutional neural network that aggregates different fine-grained images and forms image features
- Neck: A series of network layers that mix and combine image features and transmit image features to the prediction layer
- Head: The image features are predicted to generate boundary boxes and prediction categories (Fig. 1)

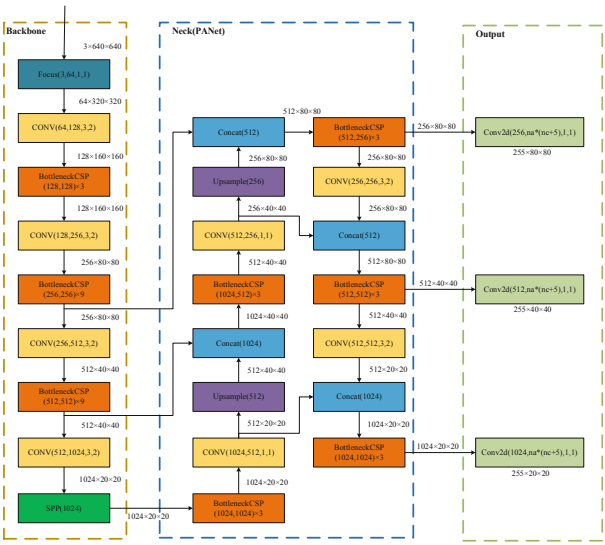


**Fig. 1.** The overall model

### 2.1 Focus

Yolov5 first uses the input of 3 × 640 × 640, and the function of the Focus layer is to copy four copies of it. Then, the four images are cut into four 3 × 320 × 320 slices through slicing operation, and then concat is used to connect the four slices in-depth, and the output is 12 × 320 × 320. After that, the output of 32 × 320 × 320 is generated through the convolution layer with the convolution kernel of 32. Finally, the output is

input to the next convolution layer through Batch Normalization and Leaky ReLU. The selection of activation functions is crucial for deep learning networks. YOLO V5s uses Leaky ReLU and Sigmoid activation functions as shown below, the middle/hidden layer uses Leaky ReLU activation functions, and the final detection layer uses Sigmoid shaped activation functions (Fig. 2).

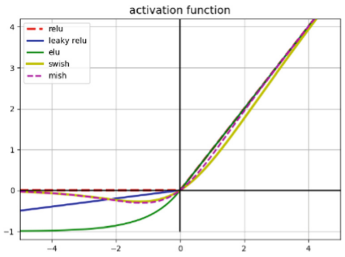


**Fig. 2.** The activation function

## 2.2 CSP

YOLO uses CSP Darknet as Backbone to extract rich information features from input images. The CSP structure of YOLO V5s is divided into two parts, Bottleneck and CSP. While bottleneck is a classic residual structure: firstly, $1 \times 1$ convolution layer (Conv + Batch Normalization + Leaky ReLU) then $3 \times 3$ convolution layer, and finally, the residual structure is added to the initial input, as shown in the following Fig. 3.

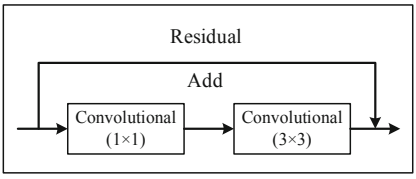


**Fig. 3.** Residual structure

## 2.3 SPP

SPP is the Spatial Pyramid Layer. It outputs the input feature graph of $512 \times 20 \times 20$ through the convolution layer of 1x1and then subsamples the three parallel convolution cores. For different branches, the padding sizes are different. In this way, the size of each pooled result is the same, and then the splicing results are added to the initial features. Finally, the 512 convolution kernel is used to restore the feature map to the size of $512 \times 20 \times 20$, as shown in the schematic diagram below (Fig. 4):

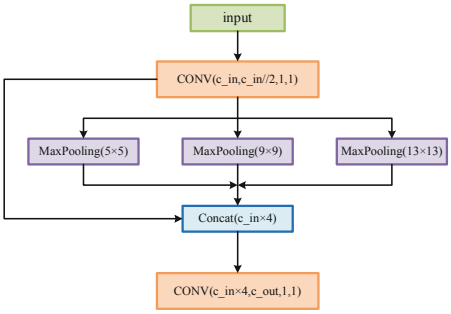


**Fig. 4.** SPP structure

## 3 Target Tracking Based on Deep SORT Algorithm

### 3.1 Motion Information

First introduced motion feature extraction part, the following eight states are described for a detection box.

- Test box center abscissa
- Longitudinal coordinates of the center of detection
- Detection frame size
- Aspect ratio
- Variable speed of the abscissa of the center of the detection box
- Variable speed of longitudinal coordinates at the center of the detection
- Variable speed of the size of the detection box
- Change speed of aspect ratio

Based on the above, use an 8-dimensional state vector to describe the change of the detection box.

$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}, \dot{r}]^T \tag{1}$$

The Kalman Filter here is a linear uniformity model [8], using a Mahalanobis distance to measure the distance between the predicted Karman filtering state and the newly obtained measurement value (Detection box), as shown below:

$$d^{(1)}(i,j) = \left(\mathbf{d}_j - \mathbf{y}_i\right)^{\mathrm{T}} \mathbf{S}_i^{-1} \left(\mathbf{d}_j - \mathbf{y}_i\right) \tag{2}$$

In the above formula, $(y_i, S_i)$ represents the projection of the $i$ -th track (Kalman Filter distribution) on the measurement space, $y_i$ is the average, and $S_i$ is the covariance, and is predicted by Kalman Filter. To perform distance measurements, must go to the same spatial distribution to do. The Mahalanobis distance calculates the uncertainty between state estimates by measuring the standard deviation and detection frame between the tracking position means of the Kalman Filter, $d^1(i,j)$ is the Mahalanobis distance between $i$-th track and the $j$-th detection, here two symbolic meaning are:

$i$: Tracking serial number
$j$: The serial number of the detection box

Using the 95% confidence interval calculated by the inverted square distribution as the threshold.

### 3.2 Appearance Information

Deep convolution network to extract the appearance characteristics of the detected target, detect and track each frame, performing a target appearance feature extraction and saving [9]. When each frame is performed later, it is necessary to perform the similarity calculation of the appearance characteristics of the current frame, the motion characteristics and appearance features will be combined as a total discrimination basis. The structure of deep neural network is shown in the following figure (Table 1):

**Table 1.** Network structure

| Name | Patch Size | Stride | Output Size |
|---|---|---|---|
| Conv 1 | 3 × 3 | 1 | 32 × 128 × 64 |
| Conv 2 | 3 × 3 | 1 | 32 × 128 × 64 |
| Max pool 3 | 3 × 3 | 2 | 32 × 64 × 32 |
| Residual 4 | 3 × 3 | 1 | 32 × 64 × 32 |
| Residual 5 | 3 × 3 | 1 | 32 × 64 × 32 |
| Residual 6 | 3 × 3 | 2 | 64 × 32 × 16 |
| Residual 7 | 3 × 3 | 1 | 64 × 32 × 16 |
| Residual 8 | 3 × 3 | 2 | 128 × 16 × 8 |
| Residual 9 | 3 × 3 | 1 | 128 × 16 × 8 |
| Dense 10 | | | 128 |
| Batch and $l_2$ normalization | | | 128 |

The network has 2,800, 864 parameters. Training the depth convolutional neural network to extract the characteristic information of the target, and trained the model on the Re-ID data set, the data set contains 1100,000 images of 1261 people, very suitable for the target tracking. On the NVIDIA GTX1050M graphics card, input 30 Bounding Box, extraction features approximately use 30 ms, and the training iterative loss value of the network is as follows (Fig. 5):

The last output of the network is a 128-dimensional vector, considering the task target tracking under 5G/6G monitoring video, the requirements for tracking algorithms are mainly pedestrian tracking, so the input size is set to 128 × 64 rectangle. Here are three definitions:

Initialization: If a detection is not associated with the previously recorded Track, then starting from this detection, initializing a new goal.

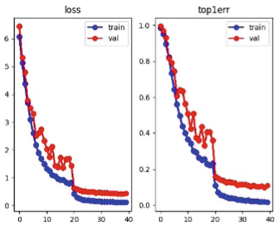


**Fig. 5.** Iterative loss value

Freshmen: After a goal is initialized, and in the first three frames are normal capture and association, then the object produces a new track, otherwise it will be deleted.
Disappearing: If the maximum save time is exceeded, if it is not associated, then this object leaves the video screen, the information of the object (the appearance features and behavior features of the record) will be deleted.

Use residual network to extract appearance features. The network accepts the target as input in the detection box, returns the vector of 128 dimensions, for each detection box (numbered $j$) inner object $d_j$, its 128 dimension of the vector is set to $r_j$, the vector of $r_j$ is 1, $\|r_j\| = 1$. A matrix is created for each target$k$, which is used to store the appearance feature (128 dimensional vector) of the target in different frames, indicated by $R_k$, the expression is as follows [10], the meaning of $k$ is the target $k$, That is, the serial number of Object-in-track, $i$ is tracking sequence number [11].

$$R_k = \left\{ r_k^{(i)} \right\}_{k=1}^{(L_k)} \tag{3}$$

The $L_k$ size is up to 100, which can only store the target appearance characteristics in 100 frames before the current time of the target $k$. At some point, you get the appearance characteristics of the detection box (numbered $j$), remember to $r_j$. The minimum cosine distance $d^{(2)}(i, j)$ is then solved by the appearance characteristics of all known $R_k$ matrices and the appearance characteristics of the obtained detection frame (numbered $j$).

$$d^{(2)}(i, j) = \min\left\{ 1 - \mathbf{r}_j^{\mathrm{T}} \mathbf{r}_k^{(i)} \middle| \mathbf{r}_k^{(i)} \in \mathcal{R}_i \right\} \tag{4}$$

## 4 Verification

With the high-definition monitoring video recorded by the camera, use the algorithm to identify and track. The effect is as shown in the figure below, it can be seen that the algorithm can again identify the target again, and give the same ID, to continue tracking.

The first case: monitoring video objectives are short-lived due to interaction (Fig. 6)

**Fig. 6.** Interaction.

The second case: surveillance video objectives have marginalized or even briefly left monitoring (Fig. 7)

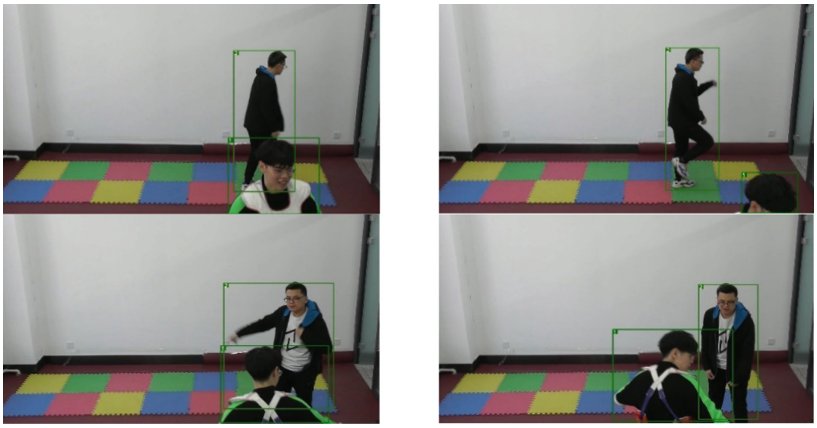

**Fig. 7.** Marginalized.

**Acknowledgment.** This work was supported by the National Natural Science Foundation of China (41861134010), the Basic scientific research project of Heilongjiang Province (KJCXZD201704), the Key Laboratory of Police Wireless Digital Communication, Ministry of Public Security (2018JYWXTX01), and partly by the Harbin research found for technological innovation (2013RFQXJ104) national education and the science program during the twelfth five-year plan (FCB150518). The authors would like to thank all the people who participated in the project.

## References


1. Wang, Z.D., Zheng, L., Liu, Y.X., et al.: Towards real-time multi-object tracking. In: 16th European Conference on Computer Vision, pp. 107–122. Springer, Heidelberg (2020)
2. Kuanhung, S., Chingte, C., Lin, J., et al.: Real-time object detection with reduced region proposal network via multi-feature concatenation. IEEE Trans. Neural Networks Learn. Syst. **31**(6), 2164–2173 (2020)
3. Luo, W.H., Xing, J.L., Milan, A., et al.: Multiple object tracking: a literature review. Artif. Intell. **293,** 103448 (2020)
4. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960); Zhang, J.W.: Gradient descent based optimization algorithms for deep learning models training [EB/OL]. [2019–03–21]. https://www.researchgate.net/publication/331670579
5. Chen, L., Ai, H.Z., Zhuang, Z.Z., et al.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: IEEE International Conference on Multimedia & Expo (ICME), pp. 1–6. IEEE, New York (2018)
6. Redmon, J.,Farhadi, A.: Yolov3: An incremental improvement [E B/O L]. [ 2 0 1 8 - 0 4 - 0 8]
7. Fu, Z.Y., Naqvi, S.M., Chambers, J.A.: Collaborative detector fusion of data-driven PHD filter for online multiple human tracking. In: Proceedings of the 21st International Conference on Information Fusion, pp. 1976–1981. IEEE, New York (2018)
8. Ren, S.Q., He, K.M., Girshick, R., et al.: Faster R-CNN: towards real- time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
9. Bewley, A., Ge, Z.Y., Ott, L., et al.: Simple online and real time tracking. In: 2016 IEEE International Conference on Image Processing, pp. 3464–3468. IEEE, New York (2016)
10. Wojke, N., Bewley, A., Paulus, D.: Simple online and real time tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing, pp. 3645–3649. IEEE, New York (2017)
11. Cipolla, R., Gal, Y., Kendall, A.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7482–7491. IEEE, New York (2018)