Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.016450
Article

Tech Science Press

HPMC: A Multi-target Tracking Algorithm for the IoT

Xinyue Lv[1], Xiaofeng Lian[2,*], Li Tan[1], Yanyan Song[1] and Chenyu Wang[3]

[1]School of Computer Science and Engineering, Beijing Technology and Business University, Beijing, 100048, China
[2]School of Artificial Intelligence, Beijing Technology and Business University, Beijing, 100048, China
[3]School of Science and Engineering, University of Dundee, Dundee, DD1 4HN, UK
[*]Corresponding Author: Xiaofeng Lian. Email:13811551604@163.com

Received: 02 January 2021; Accepted: 02 February 2021

Abstract: With the rapid development of the Internet of Things and advanced sensors, vision-based monitoring and forecasting applications have been widely used. In the context of the Internet of Things, visual devices can be regarded as network perception nodes that perform complex tasks, such as real-time monitoring of road traffic flow, target detection, and multi-target tracking. We propose the High-Performance detection and Multi-Correlation measurement algorithm (HPMC) to address the problem of target occlusion and perform trajectory correlation matching for multi-target tracking. The algorithm consists of three modules: 1) For the detection module, we proposed the You Only Look Once(YOLO)v3_plus model, which is an improvement of the YOLOv3 model. It has a multi-scale detection layer and a repulsion loss function. 2) The feature extraction module extracts appearance, movement, and shape features. A wide residual network model is established, and the coefficient k is added to extract the appearance features of the target. 3) In the multi-target tracking module, multi-correlation measures are used to fuse the three extracted features to increase the matching degree of the target track and improve the tracking performance. The experimental results show that the proposed method has better performance for small and occluded targets than comparable algorithms.

Keywords: IOT; Multi-target tracking; YOLOv3_plus; K-wide residual network model; multi-correlation measurement

1 Introduction

1.1 Research Background

In today’s life, video surveillance plays a vital role in maintaining security to control traffic, and track targets. Surveillance of video content using visual interpretation results in fatigue, missed targets, incorrect interpretation, and other problems. In contrast, intelligent video monitoring technology using artificial intelligence makes use of advanced algorithms to process massive video data, thus significantly reducing manpower, material resources, and costs and improving monitoring efficiency. The use of surveillance cameras, drones, and other Internet of Things technology provides real-time access to a large number of surveillance videos, and unmonitored areas have been significantly reduced. Researchers can develop real-time monitoring systems based on massive video data collected in real-time, such as pedestrian real-time monitoring systems that use advanced algorithms to achieve accurate positioning and tracking. Moreover, big data technology and deep learning theory [1] have transformed traditional target tracking from an inefficient method to an intelligent real-time efficient method. The detection and tracking of complex and multiple targets in surveillance video are crucial tasks in intelligent video surveillance systems. The traditional surveillance video system architecture can only provide simple functions, for example, video collection, storage, review, and query, but does not provide the ability to process intelligently the hidden information contained in the videos. In the era of rapid development of the Internet of Things, it is unrealistic to rely solely on human resources to retrieve and view massive video data. Therefore, in this study, we investigate multi-target detection and tracking based on deep learning. Deep learning has achieved remarkable success in the fields of speech recognition, natural language processing, and computer vision [2,3]. Target detection and tracking is a challenging research topic in the field of computer vision. Target detection and tracking technology has been widely used in security monitoring systems in public places such as hospitals, banks, supermarkets, and roads [4]. Deep learning has two advantages over traditional machine learning: the detection ability or classification performance is higher, and the application scope is wider for the former than the latter. A method based on deep learning does not only improve the accuracy of some algorithms but also provides functions that are difficult to achieve using traditional machine learning. Therefore, it is of great research value and significance to use deep learning technology for target detection and tracking in videos.

The crucial task in video detection and tracking is to express the content with meaningful features. Many challenges remain in video target detection and multi-target tracking due to motion blur, occlusion, morphological diversity, and illumination changes in videos. The specific problems and difficulties in video-based target detection and multi-target tracking can be summarized as follows:

- Environmental interference with target detection, such as similarity to the target, occlusion, morphological changes, and light condition changes. The key to improving the performance of video target detection is to make full use of the timing and context information of the target.
- The effect of the view of camera field. The correlation between the images acquired with different cameras is a challenging problem in tracking.
- Feature extraction. The accurate detection of similar targets requires the extraction of the features that are unique to different targets.
- Simultaneous multi-target detection and tracking. When occlusions occur between two targets, the target ID may be lost, preventing tracking. Therefore, it is crucial to maximize the degree of correlation between the target in two frames to improve the tracking and matching performance.
- Real-time tracking. Multi-target detection and tracking require an increase in the speed.

1.2 Research Content

In the paper, an association method based on high-performance detection and fusion of appearance, motion, and shape information is proposed for multi-target tracking. This paper makes contributions in three aspects:

- First, in the detection phase, the high-performance detection model You Only Look Once (YOLO) v3_plus is proposed. The method is based on the traditional YOLOv3 model, and a multi-scale detection layer and a repulsion loss function are added to improve the accuracy of small target detection and solve the occlusion problem in object detection. This idea is basically consistent with that in paper [5].
- Second, in the feature extraction stage, the appearance, motion, and shape features are extracted. A wide residual network model is established by adding the coefficient k to extract the target's

appearance features. Kalman filtering is used to extract the motion features. And the shape features are extracted using the intersection over union (IOU) value and the similarity in the width and height between the target and object of interest.

- Finally, a linear weighted fusion of the three extracted features is performed based on multiple correlations to increase the matching degree of the target and the tracking performance.

2 Related Work

Scene understanding in video data is a significant computer vision challenge. Successive detection and tracking is the preferred approach to track multiple objects, and high-quality multi-target tracking is the key task. In general, multi-target tracking algorithms divide the task into two stages: the first stage is target detection, in which the object is detected and positioned separately in each frame. The accuracy of the detection results affects the multi-target tracking performance. The second stage is tracking, in which the detected target is tracked using the formation trajectory. The tracking stage is divided into the feature extraction and fusion stages.

2.1 Target Detection

The objective of target detection is to extract the foreground or the target of interest from the video or image; the position of the target and the category of the target are determined. Real-time and accurate target detection provides good conditions for the subsequent target tracking and behavior recognition. At present, the main target detection algorithms are divided into three categories. One of the traditional target detection algorithms based on manual features is the Viola-Jones detector [6], which uses a sliding window that traverses each scale and pixel position in the image and determines whether the target face occurs in the current window. The algorithms in the second category are the target detection algorithms based on the target candidate regions. The candidate regions are extracted, and deep learning is performed on the regions to obtain the detection results. Algorithms in this category include the region-based convolutional neural network (R-CNN) [7], Fast R-CNN [8], and Faster R-CNN [9]. The algorithms in the third category are target detection algorithms based on deep learning, including YOLO [10], single-shot multibox detector (SSD) [11,12], and other methods. With the advent of deep learning, target detection algorithms have achieved breakthroughs for feature expression, time efficiency, and real-time detection.

The YOLO algorithm is a target detection method proposed by Joseph Redmon in 2016. The basic concept of this algorithm is to regard object detection as a regression problem and create spatially separated bounding boxes with class probabilities. In 2018, Redmon and Farhadi proposed the YOLOv3 [13] algorithm, which had three improvements. First, the network structure was adjusted to solve the vanishing gradient problem of the deep network. The new network structure Darknet-53 drew on the idea of ResNet [14] and added a residual network. Second, multi-scale detection was adopted to detect more fine-grained features, and three feature layers of different scales were used for target detection. Third, a logistic function was used to replace the original softmax function for predicting object categories to support multi-label objects. Based on the above analysis, YOLOv3 has high accuracy and fast speed, which is suitable for the research objectives of this study.

2.2 Target Tracking

2.2.1 Feature Extraction

Common feature extraction models include the appearance model, motion model, and composite model. The appearance model calculates object features that are easy to track; the object features encode the appearance of the object or the local area of the bounding box to track the object. The appearance model usually uses manually selected features, although they are not robust to occlusion and illumination changes in the video. The motion model encodes the motion state of the object to predict the position of the object in the subsequent frame. However, the motion model performs not well when object occlusion occurs in a sequence of many frames. Tracking based on composite models strikes a balance between appearance and motion modeling, but in practical applications, it is difficult to obtain the desired results. If a single feature is used for tracking in a complex background, the accuracy of the tracking algorithm cannot be guaranteed. Multi-feature fusion is a common method to improve tracking accuracy. In this study, the appearance, motion, and shape of the object are used to match and correlate target objects in different frames.

2.2.2 Feature Fusion

After extracting feature information, fusion is performed. Existing feature fusion methods can be categorized as multiplicative fusion and additive fusion methods. Multiplicative fusion is defined in Eq. (1), where $P(Y_j|X)$ is the probability density function of the Jth feature, Y_j is the observed value of the Jth feature, and X is the target state to be estimated.

$$P(Y_1, \ldots, Y_n|X) = \prod_{j=1}^{n} P(Y_j|X) \tag{1}$$

In the additive fusion method, it is assumed that the target state is given. The corresponding weight value of a feature is assigned, and the combined observed likelihood value of n features after fusion is obtained using a weighted summation. Additive fusion is defined in Eq. (2), where ω^j is the weight value of the observed value of the Jth feature, $\sum_{j=1}^{\mathrm{n}} \omega^j = 1$.

$$P(Y_1, \ldots, Y_n|X) = \prod_{j=1}^{n} \omega^{\mathrm{j}} P(Y_j|X) \tag{2}$$

Although multiplicative fusion is straightforward, it assumes that the features are independent, whereas additive fusion does not require independent features and is insensitive to noise. Therefore, the linear weighting method is used in this study to fuse multiple features.

2.2.3 Mainstream Target Tracking Algorithms

Current mainstream target tracking algorithms include Kalman filtering [15], which is regarded as one of the best Bayesian filtering methods when target tracking occurs under ideal conditions (linear, Gaussian stationary). In recent years, researchers have also proposed improvements based on particle filter methods, such as the boosted particle filter (BPF) [16], visual tracking decomposition (VTD) [17], and a particle filter with a Markov Chain Monte Carlo (MCMC) sampling step [18]. Huang [19] proposed an improved KCF-based robust tracking algorithm, which solves the problems of sensitivity to illumination, scale changes and occlusion in the Kernel Correlation Filter tracker. Bewley [20] proposed a sort algorithm that propagates the state of the tracking object into the future frames (using Kalman filtering and the assumption of linear speed) and associates the current detection object with existing objects. Wojke [21] proposed the Deepsort algorithm in which a neural network module was added to identify pedestrians. This approach prevented the ID loss in case of occlusion. The tracking model in this study is improved using the Deepsort algorithm, which fuses the appearance, movement, and shape information to obtain multiple correlations to improve the tracking performance.

3 Method

We propose the High-Performance detection and Multi-Correlation measurement (HPMC) algorithm to address the problem of target occlusion and determine the correlation between the target in different frames. The structure of the model is shown in Fig. 1. The target detection module is based on a high-performance detection method, namely the improved YOLOv3 target detection algorithm YOLOv3_plus. In the tracking module, the appearance, movement, and shape of the target are used for tracking.

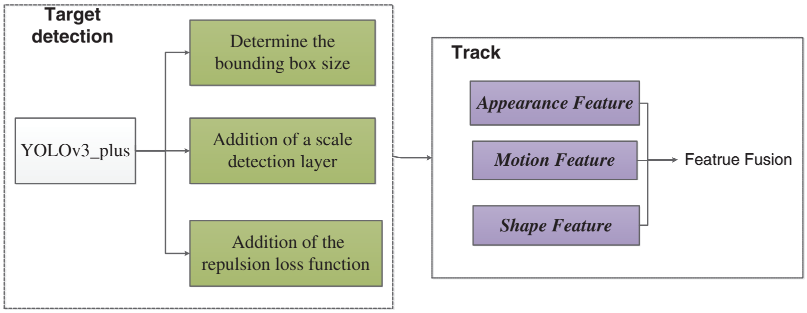
Target detection
YOLOv3_plus
Determine the bounding box size
Addition of a scale detection layer
Addition of the repulsion loss function
Track
Appearance Feature
Motion Feature
Shape Feature
Featrue Fusion

Figure 1: The structure of the multi-target tracking model

3.1 Target Detection

The multi-target tracking method is divided into two stages. The first stage is the target detection in the video, and the second stage is the correlation of the detection results. The YOLOv3 algorithm has the disadvantages of low accuracy for object location and a low recall rate. Therefore, we propose the new detection model YOLOv3_plus. The details of the model are described in this section.

3.1.1 Determine the Bounding Box Size

First, cluster analysis is performed on the data set to determine the size of the bounding box. Eq. (3) is used to measure the distance between the candidate boxes. The data set is divided into K clusters according to the distance between the borders of the boxes. The distance within the cluster is kept as small as possible through iteration, whereas the distance between the clusters is as large as possible. We determine the size of the candidate box by changing the value of the target function.

$$D(S_b, S_c) = 1 - IOU(S_b, S_c) \tag{3}$$

where S_b represents the set of real boxes, S_c represents the center of the cluster of the bounding boxes, and $IOU(S_b, S_c)$ represents the ratio of the intersection to the union between the real box and the cluster center of the bounding boxes. The larger the IOU value, the higher the correlation is between the two, or the higher the similarity is.

3.1.2 Addition of A Scale Detection Layer

A 104 × 104 prediction layer was added to the original network to solve the problem of inaccurate detection of small targets. Shallow features can be easily located in a small-scale target, but the semantic information is weak. Deep features contain rich semantic information, but the location information of a small-scale target is difficult to obtain. Therefore, residual information is used to fuse the information on shallow and deep features for target detection. However, it is not possible to use the deep-feature map for the semantic enhancement of shallow features; therefore, when the fusion layer is created, the deep-feature map is enlarged to the same size as the shallow-feature map for subsequent fusion connection using up-sampling. The structure diagram of the network model with the added fusion detection layer is shown in Fig. 2.

As shown in Fig. 2, the 104 × 104 scale detection layer is added to the original YOLOv3 network structure. The image is divided into a fine grid to detect smaller objects and significantly improve the detection performance for small targets.

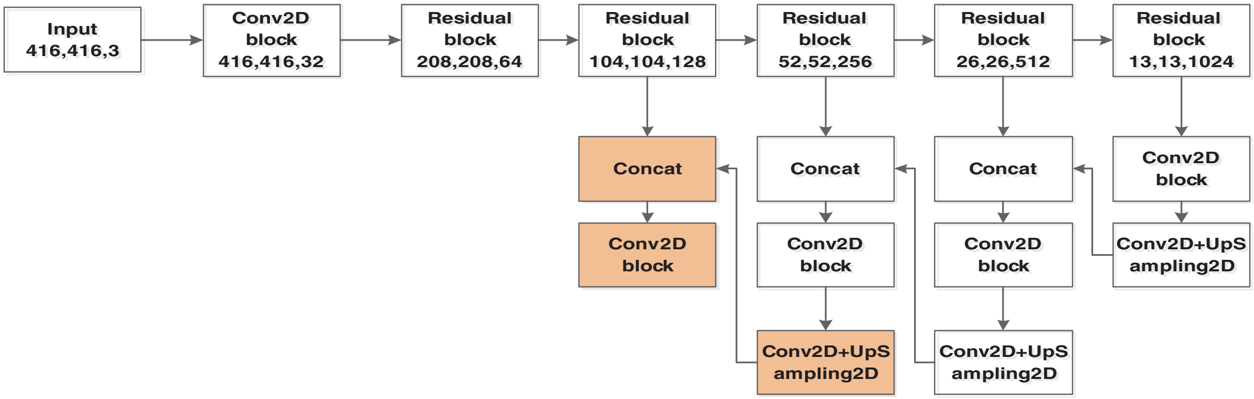

Figure 2: The network structure diagram with the added fusion detection layer

3.1.3 Addition of the Repulsion Loss Function

The repulsion loss function is added to the YOLOv3 algorithm to increase the distance between the prediction box and the surrounding non-labeled boxes, which prevents omissions due to target occlusion. The updated loss function ensures that the prediction box is close to the target, thus reducing the detection error of the model. The loss function is defined in Eq. (4).

$$\begin{cases} IoG(B,G)\dfrac{area(B\cap G)}{area(G)} \\ Smooth_{ln}=\begin{cases} -\ln(1-x), & x\le\sigma \\ \dfrac{x-\sigma}{1-\sigma}-\ln(1-\sigma), & x>\sigma \end{cases}\sigma\in[0,1) \\ L_{PBox-GBox}=\dfrac{\sum_{P\in P_+} Smooth_{ln}(IoG(B^P,G^P_{GBox}))}{|P_+|} \\ L_{PBox-PBox}=\dfrac{\sum_{i\neq j} Smooth_{ln}(IoU(B^{P_i},B^{P_j}))}{\sum_{i\neq j}\mathrm{I}[IoU(B^{P_i},B^{P_j})>0]+\varepsilon} \\ Loss=L_{yolov3}+\alpha * L_{PBox-GBox}+\beta * L_{PBox-PBox} \end{cases} \tag{4}$$

where L_{yolov3} represents the loss calculation value of the prediction box and the label box in the regression. The center point, length, width, category, and confidence value of the grids of YOLOv3 are obtained; $L_{PBox-GBox}$ represents the calculated value of the offset between the prediction box and the labeled labeling boxes; $L_{PBox-PBox}$ represents the calculated value of the loss near the prediction box and the other prediction boxes, and α, β is used to balance the weight of the two loss values. P_+ represents the prediction box set, B^P represents the prediction box corresponding to the candidate box, G^P_{GBox} represents the labeled box with the largest IOU area, except for the real labeled box, I represents the distance between the prediction boxes $IOU > 0$; ε is a very small constant.

Therefore, the YOLOv3_plus detection model can be used to conduct target detection on the dataset to obtain high-performance detection results to improve the multi-target tracking performance. The structure of the YOLOv3_plus detection model is shown in Fig. 3.

3.2 Feature Extraction

The relationship between the detected targets, namely the data association, is established in different frames to obtain the target trajectory. Occlusion between objects and similar appearances will increase the difficulty of data association. Therefore, the object's appearance, movement, and shape are determined, and the correlation between the objects in different frames is determined, as shown in Fig. 4.

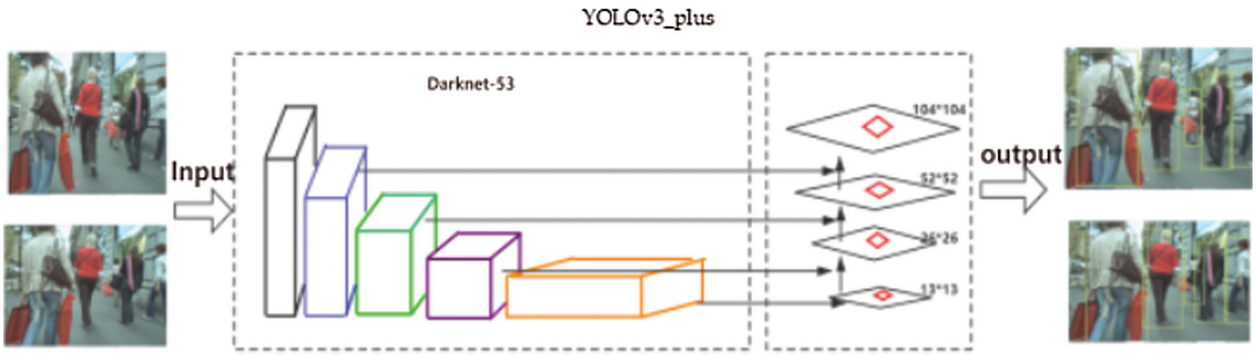

Figure 3: The structure of the YOLOv3_plus detection model

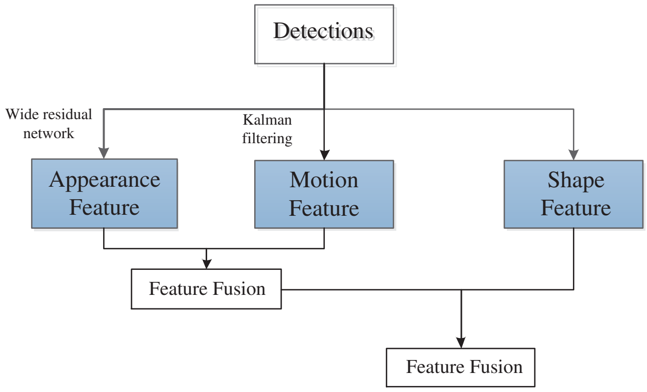

Figure 4: Multi-correlation measurement

3.2.1 *Appearance Feature*

The feature vectors are extracted from the objects detected in the video frames using the wide residual network. The distance between the detection target and the feature vectors of the target contained in the track is used to determine the degree of matching degree between the detected object and the track, i.e., the correlation measure of the appearance feature, which is called A_f.

The CNN with a residual structure in the DEEPSORT algorithm is used to extract the appearance of the detection target. The similarity in the appearance of the targets between different frames is calculated to obtain the data association, and the trajectory is determined. The network structure is shown in Fig. 5.

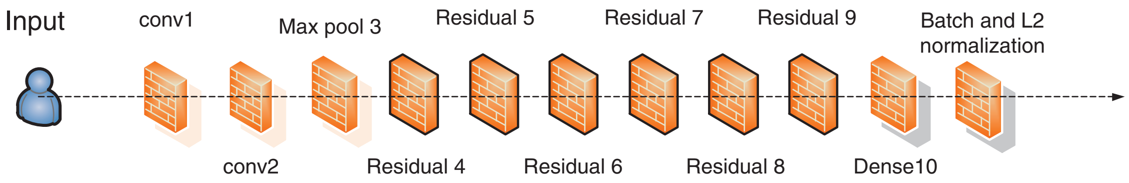

Figure 5: Appearance measurements

A wide residual network structure [22] has a better capability of determining the parameter values of the expected function than a simple multi-layer network. In addition, gradient disappearance is prevented in the optimization of the network for a large number of layers. Therefore, the coefficient k is added to the original residual module in this study to increase the number of convolution kernels and form a wide residual network. As shown in Fig. 6, the left side is the original residual network, and the right side is the wide residual network. This approach reduces the number of layers and speeds up the calculation.

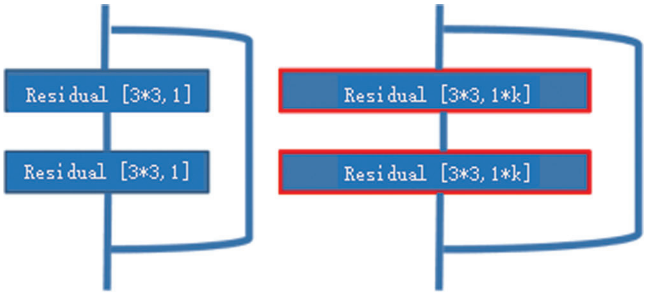

Figure 6: Wide residual network

3.2.2 Motion Feature

Kalman filtering [23] adopts the state-space model of noise and signal and uses the estimated value of the previous moment and the observed value of the current moment to update the estimation of the state variables and obtain the estimated value of the current moment. This method is suitable for real-time processing. Therefore, Kalman filtering is used to obtain the motion information of the detection targets and predict the inter-frame displacement of the targets. The observed state of the target includes the position and velocity information, and the state of each target can be modeled, as defined in Eq. (5). where p_t is the position state of the target, and v_t is the velocity state of the target.

$$x_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix} \tag{5}$$

When the acceleration a_t and the adjacent time interval Δt are known, v_t can be expressed as shown in Eq. (6), and p_t can be expressed as shown in Eq. (7).

$$v_t = v_{t-1} + a_t \cdot \Delta t \tag{6}$$

$$p_t = p_{t-1} + v_{t-1} \cdot \Delta t + a_t \cdot \frac{\Delta t^2}{2} \tag{7}$$

The state prediction of Kalman filtering is defined in Eq. (8).

$$\hat{x}_t^- = F_t \hat{x}_{t-1} + B_t a_t \tag{8}$$

where, $F_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ represents the state-transition matrix, and $B_t = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$ represents the control matrix. This variable $\hat{x}_t^-$ is an estimator that is predicted based on the state in the last moment. The variable is affected by noise. The higher the noise level, the greater the uncertainty of the predicted state is; therefore, correction is required. A covariance matrix is commonly used to predict the state. We define the uncertainty as the changes of the noise covariance matrix, which is defined in Eq. (9).

$$P_t^- = FP_{t-1}F^T + Q \tag{9}$$

The noise covariance matrix in the current state is inferred from the noise covariance matrix in the previous moment. Q represents the noise covariance matrix of the prediction model.

If a discrepancy exists between the predicted state value and the actual value, it is necessary to update the predicted state. If the observed value is Z_t, it can be expressed as Eq. (10):

$$Z_t = Hx_t + u \tag{10}$$

where $H = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ is the observation matrix; only the position in the movement state of the detection target is observed. u represents the noise introduced in the observation process, and its covariance matrix is R. When the filter is implemented, the measured noise covariance R is observed; this is the known condition of the filter. Finally, the updated equation of the predicted state is Eq. (11).

$$\begin{cases} K_t = \dfrac{P_{t-1}H^T}{HP_t{}^- H^T + R} \\ \hat{x}_t = \hat{x}^- + K_t(Z_t - H\hat{x}^-) \end{cases} \tag{11}$$

where K_t represents the Kalman gain, which is used to balance the size of the covariance matrix P in the prediction state and the noise covariance matrix R introduced in the observation and determine whether the final updated correction is biased toward the prediction model or the observation model according to the size of P and R. In addition to updating the predicted value of the current state, it is also necessary to update the noise covariance matrix as follows:

$$P_t = (I - K_t H)P_t{}^- \tag{12}$$

The target region in the next frame of the detection target is predicted by using the defined Kalman filtering. In the next frame, target matching is performed in the prediction region. The distance between the detection target and the trajectory predicted by the Kalman filtering is used to describe the degree of motion matching, i.e., the correlation measure of the motion feature, called M_f.

3.2.3 Shape Feature

The shape information of the detection target and trajectory is the third feature that is calculated, i.e., the correlation measure of the shape feature, which is called S_f. This feature compensates for the changes occurring between frames or partial occlusion and is calculated as follows:

$$\begin{cases} d^{(1)} = e^{-\left(\dfrac{\left|H_{trk_i} - H_{det_j}\right|}{H_{trk_i} + H_{det_j}} + \dfrac{\left|W_{trk_i} - W_{det_j}\right|}{W_{trk_i} + W_{det_j}}\right)} \\ d^{(2)} = 1 - IOU(trk_i, det_j) \\ cost = \lambda d^{(1)} + (1 - \lambda)d^{(2)} \end{cases} \tag{13}$$

where trk_i represents the trajectory i, det_j represents the current detection target j, H_{trk_i}, W_{trk_i} represents the width and height of the trajectory, H_{det_j}, W_{det_j} represents the width and height of the detection target j, $IOU(\cdot)$ represents the IOU value of the two bounding boxes, and λ is the weight coefficient.

3.3 Feature Fusion

The appearance features A_f and motion features M_f are combined linearly using weights, and the fusion result is denoted as f1. f1 is then combined with the shape features S_f linearly using weights to obtain the fusion result f, as shown in Eq. (14).

$$\begin{cases} \mathrm{f1} = \lambda_1 A_f + (1 - \lambda_1)\mathrm{M_f} \\ \mathrm{f} = \lambda_2 \mathrm{S}_f + (1 - \lambda_2)\mathrm{f1} \end{cases} \tag{14}$$

3.4 *Evaluation Indices*

Evaluation indices were used to determine the accuracy of the proposed multi-target tracking algorithm regarding the number of detected targets and the state and trajectory of the target, i.e., to retain the ID of the target in subsequent frames. The classification of activities, events, and relationships–multiple object tracking precision (CLEAR MOT) index [24] was selected; the multiple object tracking accuracy (MOTA) is defined in Eq. (15). The *MOTP* is defined in Eq. (16).

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{15}$$

where *FN* represents the total number of missed targets, *FP* represents the total number of false positives, *IDSW*represents the total number of ID switches, and *GT* represents the number of correctly labeled objects. *MOTA* considers the number of errors in object matching during tracking in all frames. It is an intuitive measure to determine the performance of a tracker in detecting objects and maintaining the track and is independent of the estimation accuracy of the object position.

$$MOTP = \frac{\sum_{t,i} d_t^i}{\sum_t c_t} \tag{16}$$

where *d* represents the distance between the detection result of the matched detector and the output of the tracker, and *c* represents the matched logarithm of the detection result of the detector and the output of the tracker. *MOTP* represents the ability of the tracker to estimate the precise target position independent of the identification of the target configuration, and the ability to maintain a consistent trajectory. The definitions of the other evaluation indices are as follows:

IDF1: The ratio of the number of targets correctly identified to the average of actual targets and calculated targets. Rcll: The ratio of the number of correctly matched detection targets to the number of labeled targets. IDP: The test score for the correct identification. IDR: Correctly identify the test score of Ground Truth. The higher the index value, the better the tracking performance is.

FN: The total number of false negatives. FP: The total number of false positives. FM: The number of times that tracking is interrupted, i.e., where the track does not match the actual track. IDSW: The total number of ID switches. The lower the index value, the better the tracking performance is.

4 Experiment Results and Discussion

The experimental platform was a DELL server PowerEdge R730; operating system: Ubuntu 14.04, GPU: NVIDIA Tesla K40m × 2, video memory: 12GB × 2, CPU: Intel (R) Core i3 3220, memory 64 GB. A Pytorch framework was used to implement and evaluate the proposed HPMC algorithm. The parameter settings are shown in Tab. 1.

Table 1: Parameter settings

Variable name	Value	help
min_confidence	0.3	Detection confidence threshold
nms_max_overlap	1.0	Non-maxima suppression threshold: Maximum
min_detection_height	0	Threshold for the detection bounding box
max_cosine_distance	0.2	Gating threshold for cosine distance
nn_budget	100	Maximum size of the appearance descriptors, i.e., the number of frames for tracking.

The tracking results (CLEAR MOT index) of the proposed tracking model are shown in Tab. 2 for the 7 videos in the MOT16 dataset [25]. As shown in Tab. 2 and Fig. 7, the proposed tracking model has a high accuracy for the identification of the track. The results show that high-performance detection minimizes the drift of the trajectory and improves the tracking performance. In addition, the integration of the appearance, movement, and shape features minimizes the occlusion error that can affect the tracking results in a dense crowd. As shown in Fig. 7, the MOTA of the proposed HPMC model is higher than 45% for the seven videos, and the MOTP is around 20%. Fig. 8 shows that the tracking accuracy of the HPMC model in all 7 videos is 59.7%. Therefore, the proposed HPMC model has a high tracking accuracy not only for all datasets of MOT16 but also for the sub-samples.

Table 2: Results of the CLEAR MOT index of the HPMC model

	IDF1	IDP	IDR	Rcll	FP	FN	FM	IDSW
1	47.6%	70.8%	35.8%	48.1%	443	9247	183	89
2	73.1%	85.7%	63.8%	69.3%	2460	14613	182	51
3	62.5%	80.7%	51.0%	57.9%	359	2872	94	38
4	61.4%	73.2%	52.8%	69.1%	165	1626	55	28
5	62.6%	70.7%	56.2%	71.4%	993	3519	278	116
6	64.6%	74.4%	57.1%	72.4%	397	2534	74	31
7	60.6%	68.0%	54.6%	69.4%	1255	3502	308	184
All	65.0%	78.1%	55.6%	65.7%	6072	37913	1174	537

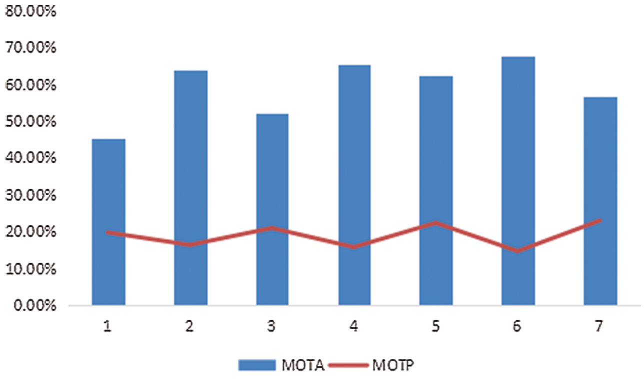

Figure 7: Results of the MOTA and MOTP for the seven videos

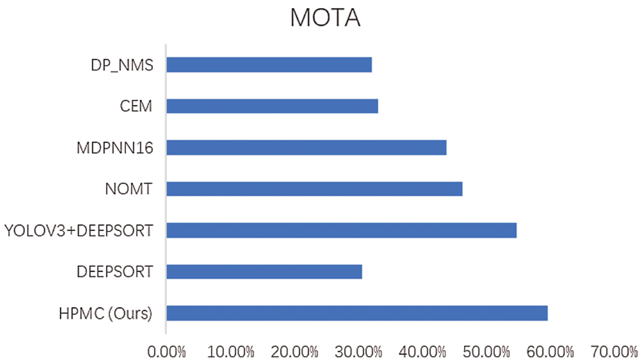

Figure 8: Results of the MOTA for the different tracking models

The comparison of the results of the proposed HPMC, the DEEPSORT tracking model and other mainstream algorithms on the MOT16 dataset is shown in Tab. 3.

Table 3: Results of the CLEAR MOT index of the different tracking models

	HPMC (Ours)	DEEPSORT	YOLOV3 +DEEPSORT	NOMT [26]	MDPNN16 [27]	CEM [28]	DP_NMS [29]
FP	6072	5873	11773	9753	3501	6837	1123
FN	37913	70261	37586	87565	98193	114322	121579
IDSW	537	352	537	359	723	642	972

Tab. 3 shows that the HPMC model provides better performance than the DEEPSORT tracking model. As shown in Fig. 8, the accuracy is 59.7% for the HPMC model, 54.8% for YOLOV3+DEEPSORT, and 30.7% for the DEEPSORT tracking model. The accuracy of HPMC is higher than other tracking models. However, the tracking accuracy has been reduced, and the accuracy of the position estimation of the people requires improvement.

Examples of the video frames are shown in Figs. 9 and 10 to illustrate the results. The location and ID of the proposed tracking model are marked in the image. Each target in the figure has two annotations. The red box indicates the detection target of the current frame, and the border with the ID information is the target track of the previous frame. Fig. 9 shows the results for a target with occlusion, and Fig. 10 shows the results for multiple small targets.

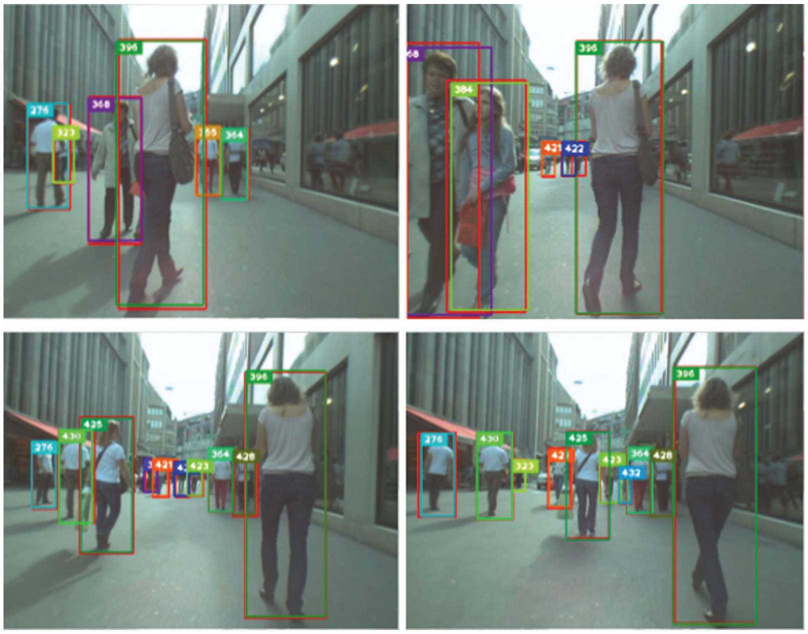

Figure 9: Examples of the tracking results for a target with occlusion

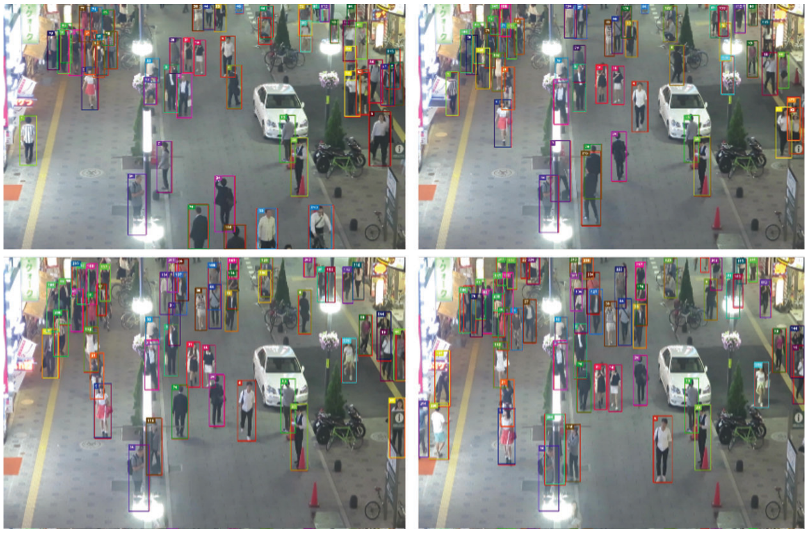

Figure 10: Examples of the tracking results for multiple small targets

5 Conclusions

With the development of sensor technology and the Internet of Things, an accurate and effective target tracking method suitable for video data is required. In this study, the multi-target tracking model HPMC is proposed to address the problems of long-term drift in tracking, in-category occlusion, and similarity in appearance of the target. The HPMC model integrates the appearance, motion, and shape features using correlation measures. The algorithm consists of three modules: 1) a detection module that is based on the YOLOv3 model with a multi-scale detection layer and repulsion loss function. 2) A feature extraction module is used to extract the appearance, movement, and shape features. A wide residual network model is established, and the coefficient k is used to extract the appearance features of the target. 3) The multi-target tracking module uses the multi-correlation measures to fuse the three extracted features to increase the matching degree of the target track and improve the tracking performance. The experimental results on the dataset MOT16 showed that the proposed HPMC model had higher accuracy than the DEEPSORT algorithm and the combination of the YOLOv3 and DEEPSORT algorithm. The HPMC model is well suited for the detection of small and occluded targets.

Funding Statement: This research was funded by Beijing Municipal Natural Science Foundation-Haidian original innovation joint fund (L182007); the National Natural Science Foundation of China (61702020) and supporting projects (PXM2018_014213_000033).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Han and C. Zhang, “Big data analysis on economical urban traffic in Beijing: Organize overlapping transportation though the underground diameter line of Beijing railway hub,” in *Proc. ICCCBDA*, ChengDu, China, pp. 269–273, 2019.

[2] W. Fang, F. Zhang, Y. Ding and J. Sheng, “A new sequential image prediction method based on LSTM and DCGAN,” *Computers, Materials & Continua*, vol. 64, no. 1, pp. 217–231, 2020.

[3] W. Fang, L. Pang and W. N. Yi, “Survey on the application of deep reinforcement learning in image processing,” *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 39–58, 2020.

[4] F. Bi, X. Ma, W. Chen, W. Fang, H. Chen *et al.,* “Review on video object tracking based on deep learning,” *Journal of New Media*, vol. 1, no. 2, pp. 63–74, 2019.

[5] Y. Y. Song, L. Tan, Z. H. Ma and X. P. Ren, “Video target detection based on improved YOLOV3 algorithm,” *Journal of Frontiers of Computer Science and Technology*, vol. 15, no. 1, pp. 163–172, 2020.

[6] P. Viola and M. J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[7] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for object detection and semantic segmentation,” in *Proc. CVPR*, Columbus, USA, pp. 580–587, 2014.

[8] R. Girshick, “Fast R-CNN,” in *Proc. ICCV*, Santiago, Chile, pp. 1440–1448, 2015.

[9] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.

[10] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. CVPR*, Las Vegas, NV, USA, pp. 779–788, 2016.

[11] W. Liu, D. Angueliv, D. Erhan, C. Szegedy, S. Reed *et al.,* “SSD: Single shot multiBox detector,” in *Proc. ECCV*, Amsterdam, Netherlands, pp. 21–37, 2016.

[12] G. H. Yu, H. H. Fan, H. Y. Zhou, T. Wu and H. J. Zhu, “Vehicle target detection method based on improved SSD model,” *Journal on Artificial Intelligence*, vol. 2, no. 3, pp. 125–135, 2020.

[13] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” in *Proc. CVPR*, Salt Lake City, Utah, USA, 2018. Available: https://arxiv.org/abs/1804.02767

[14] K. M. He, X. Y. Zhang, S. Q. Ren and J. Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, Las Vegas, Nevada, USA, pp. 770–778, 2016.

[15] H. M. Wang, L. L. Huo and J. Zhang, “Target tracking algorithm based on dynamic template and Kalman filter,” in *Proc. ICCSN*, Xi’an, China, pp. 330–333, 2011.

[16] K. Okuma, A. Taleghani, N. D. Freitas, J. J. Little and D. G. Lowe, “A boosted particle filter: Multitarget detection and tracking,” in *Proc. ECCV*, Prague, Czech Republic, pp. 28–39, 2004.

[17] J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *Proc. CVPR*, San Francisco, California, USA, pp. 1269–1276, 2010.

[18] Z. Khan, T. Balch and F. Dellaert, “MCMC-based particle filtering for tracking a variable number of interacting targets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.

[19] D. Huang, P. Gu, H. Feng, Y. Lin and L. Zheng, “Robust visual tracking models designs through kernelized correlation filters,” *Intelligent Automation & Soft Computing*, vol. 26, no. 2, pp. 313–322, 2020.

[20] A. Bewley, Z. Y. Ge, L. Ott, F. Ramos and B. Upcroft, “Simple online and realtime tracking,” in *Proc. ICIP*, Phoenix, Arizona, USA, pp. 3464–3468, 2016.

[21] N. Wojke, A. Bewley and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. ICIP*, Beijing, China, pp. 3645–3649, 2017.

[22] S. Zagoruyko and N. Komodakis, “Wide residual networks,” in *Proc. BMVC*, York, UK, pp. 87.1–87.12, 2016.

[23] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[24] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” *Eurasip Journal on Image & Video Processing*, vol. 2008, no. 1, pp. 246–309, 2008.

[25] A. Milan, L. Leal-Taixe, I. Reid, S. Roth and K. Schindler, “MOT16: A benchmark for multi-object tracking,” 2016. [Online]. Available: https://arxiv.org/abs/1603.00831.

[26] W. G. Choi, “Near-online multi-target tracking with aggregated local flow descriptor,” in *Proc. ICCV*, Santiago, Chile, pp. 3029–3037, 2015.

[27] A. Sadeghian, A. Alahi and S. Savarese, “Tracking the untrackable: Learning to track multiple cues with long-term dependencies,” in *Proc. ICCV*, Venice, Italy, pp. 300–311, 2017.

[28] A. Milan, S. Roth and K. Schindler, “Continuous energy minimization for multitarget tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, 2014.

[29] H. Pirsiavash, D. Ramanan and C. C. Fowlkes, “Globally optimal greedy algorithms for tracking a variable number of objects,” in *Proc. CVPR*, Colorado Springs, USA, pp. 1201–1208, 2011.