Improving Online Multi-Person Tracking Occlusion: Scale Loss for Deep ReID Feature Learning

by

Hongyu Yang

to obtain the degree of Master Science
in Computer Science
at the Delft University of Technology,
to be defended publicly on Thursday November 29, 2018 at 10:00 AM.

Student number:	4718763	
Project duration:	March 5, 2018 – November 29, 2018	
Thesis committee:	Prof. dr. M. J. T. Reinders,	TU Delft, supervisor
	Dr. J. C. van Gemert,	TU Delft daily supervisor
	Dr. Anna Villanova,	TU Delft
	Dr. Sicco Verwer,	TU Delft

This thesis is confidential and cannot be made public until November 28, 2018.

TU Delft

Improving Online Multi-Person Tracking Occlusion: Scale Loss for Deep ReID Feature Learning

Hongyu Yang
Delft University of Technology
Mekelweg 5, 2628 CD Delft
H.Yang-10@student.tudelft.nl

Abstract

Occlusion and crossing in Multi-Person Tracking always influence the tracking results. In this paper, we show how deep Re-Identification (ReID), which aims at matching pedestrians across non-overlapping video cameras, can be used to improve the occlusion problem on tracking. The learned ReID feature is more robust than other features used in traditional trackers because the training set is collected from different cameras which includes different parts of the same person. This also helps to solve the occlusion problem in tracking. We train a neural network with the designed scale loss which normalizes both weight vectors and output features to remove the effect of their scale variations on a large Person ReID dataset offline to learn the deep ReID model and build a framework combining detector and tracker to meet real-world application requirements. During the online tracking stage, the data association is solved by calculating the cosine distance cost matrix according to the learned ReID feature vectors. Experiments show that using ReID features can effectively reduce the occlusion index data on MOTChallenge, and the scale loss performs well. Overall our method achieves competitive performance on MOTChallenge, and the framework guarantees the running speed in real-time.

1. Introduction

Object Tracking has received increasing attention in Computer Vision due to its academic and commercial potential. It is the basis of some high-level task, such as behavior analysis and motion recognition and at the same time, it is widely used in video surveillance, human-computer interaction, virtual reality, and medical imaging. Object Tracking is divided into two sub-topics: Single-Object Tracking and Multi-Object Tracking. The Single-Object Tracking is through the object's apparent modeling or motion modeling to deal with lighting, deformation, occlusion and other issues. In addition to the problems encountered by Single-Object Tracking, Multi-Object Tracking requires association matching between objects. In the Multi-Object Tracking task, frequent occlusion of the target influenced the performance of the tracker. Occlusion and crossing have always been difficult points for Multi-Object Tracking. When the target is deformed due to these two kinds of problems, the traditional tracking algorithm is complicated to identify the correct target, as shown in Figure 1.

(a) Correlation filter - Dlib tracker[1]

(b) Kalman filter traker[2]

Figure 1. When two person crossing happens, traditional tracking methods cannot recognize the correct target. In the figures, the number 1, 2 and 3 stands for the identification result of a person obtained by trackers. The detector used here is MaskRCNN[3]

To solve the occlusion and crossing problem in MOT, we propose an online approach which is evaluated on the MOTChallenge dataset [4, 5]. A framework combining detector and tracker is designed to meet the practical application. This paper is to deal with *Multi-Person* Tracking problem, where the "Object" in MOT denotes "Person" in our case.

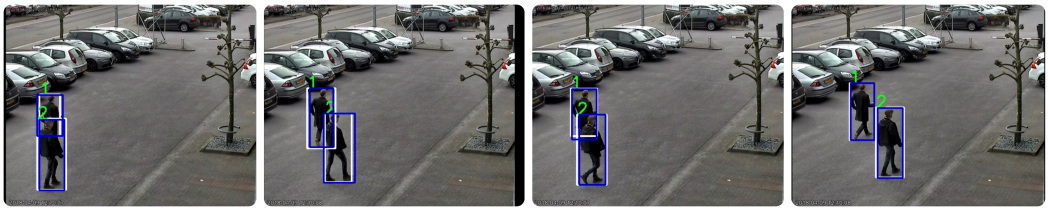

Figure 2. Person Re-Identification (ReID) improved the occlusion and crossing problem. The blue bounding boxes are detection results, the white bonding boxes and corresponding number on top of them are tracking results. The detector used here is still MaskRCNN

There is another subclass in computer vision: Person Re-Identification (Reid). ReID is to determine whether a person in a camera has ever appeared in other cameras by computing the distance between features of two images that same class images have smaller distance while different class images have more considerable distance. The training set of ReID contains different body parts of the same person, and under different cameras, the same person will show different perspectives. During Multi-Person Tracking, when the occlusion or crossing happens, part of the persons' bodies is overlapping. In other words, we believe that an effective ReID model can handle the occlusion or crossing problem since the trained network is robust to deal with the "identity" problem of the targets, which motivates us to apply ReID method in the assignment matching part of tracking to solve the occlusion and crossing problems. In particular, a scale loss, which normalizes both weight vectors and output features to remove the effect of scale variations, is applied to learn more discriminative deep features which are vital to improving the ReID performance. We evaluated our research hypothesis on the MOTChallenge dataset.

We have the following contributions: 1) Designing a real-time tracking framework which combined ReID with a standard tracker to solve the occlusion and crossing problem, 2) A scale loss is applied to learn more discriminative feature representations to improve the ReID performance. The rest of the paper is organized as follows: section 2 describes the related work, section 3 introduces the proposed SODR Tracker, section 4 evaluates the method against the publicly available MOTChallenge dataset, section 5 is the conclusions.

2. Related Work

Multi-Object Tracking (MOT) can be divided into online tracking and offline tracking methods. The difference is whether the target of the last few frames is used when processing the current frame. In online tracking[6, 7], the image sequence is frame by frame. The tracking method is therefore also called sequence tracking. Offline Tracking[8, 9] uses a set of frames to process data. The observation targets from all frames need to be acquired in advance and then analyzed to calculate the final output. In this paper, we consider online tracking methods.

Some online approaches use a motion model to capture the dynamic behavior of a target, which estimates the potential location of the target in future frames, thereby reducing the search space. The linear motion model is currently the most mainstream model which assumes the target move with an average speed[10]. The nonlinear motion model can solve more complicated situations. It makes the motion similarity between tracks more accurate. For example, yang *et al*. [11] uses the nonlinear motion model to deal with the problem of free movement of the target. In addition to the motion model, there are other methods[12] using the target as a Gaussian distribution in the image space, and then explicitly occluding the occlusion rate of all target pairs in the form of a partial energy difference function. Such probabilistic prediction methods usually use the target state as an uncertain distribution. The algorithm only needs past or present observation targets, so it is also particularly suitable for online tracking. A variety of probabilistic prediction models are used in Multi-Object Tracking, such as Kalman filter[13, 14], extended Kalman filter[15], and particle filter[16]. In our method, we used a Kalman filter to predict the motion track of targets.

Appearance model is the most important way to calculate the similarity between detected results and tracking results in MOT. Some scholars use local features. After obtaining these features, they can be used to generate short trajectories[17], estimated camera motion[18], motion clustering[19] and so on. The optical flow method can also be considered as local features. When we use the pixel unit as the best local range, many MOT methods use the optical flow method to generate short tracks before data association[20, 21]. Some scholars also use region features. Compared to local features, the region features to search for a wider range of bounding boxes. The most commonly used representation methods are classic color histograms[15] and raw pixel templates[22]. Color histograms are often used. However, they ignore the spatial distribution of the target area. Local features are efficient but sensitive to occlusion and out-of-plane. Gradient-based features such as HOG can describe the shape of the target and are adaptable to certain

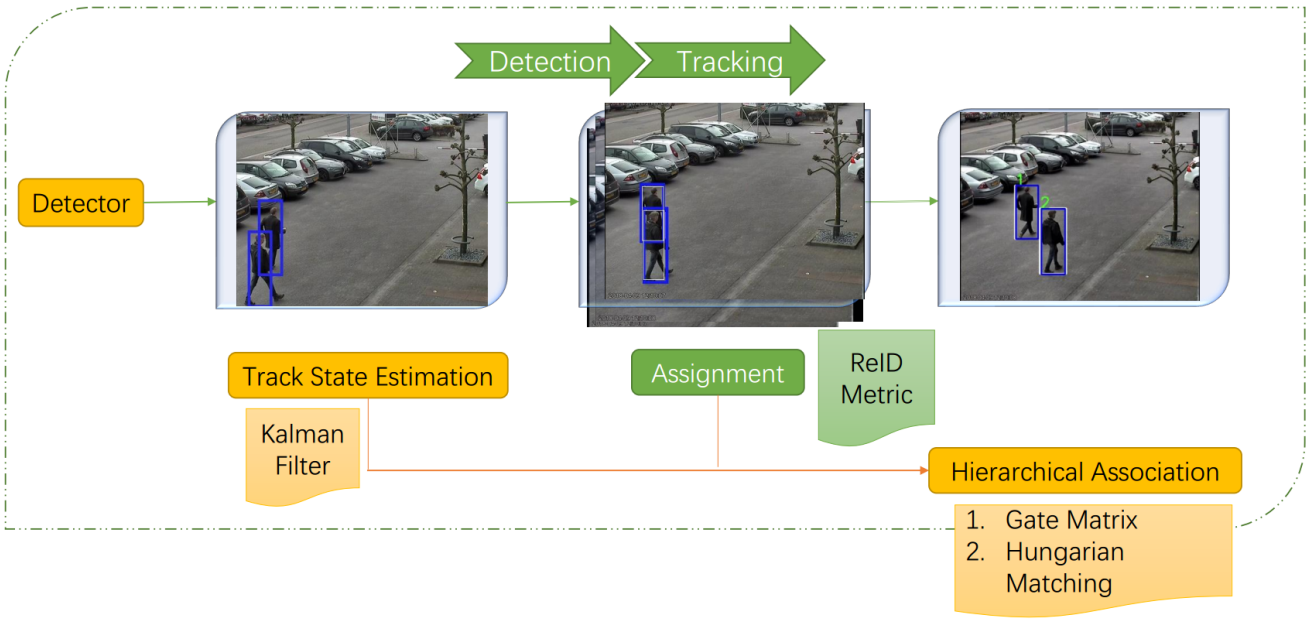

Figure 3. The green parts are our contribution and the orange parts we use others work. This framework is to combine detection and tracking. The blue bounding boxes are detection results, the white bonding boxes and corresponding number on top of them are tracking results.

changes such as illumination, but it does not handle occlusion and deformation well. Regional covariance matrices are relatively robust because they use more information, but at the same time bring higher computational complexity. Some of the depth features make the calculation of similarity more accurate, but it requires multi-view information of the same scene or an additional algorithm[23] to obtain the depth. These features are not suitable to the problem we are going to solve thus we did not use them in our method.

Recently, many scholars have tried to use deep learning methods to assist in appearance modeling and have achieved good results in the MOT competition[24, 25]. Leal *et al*. [26] trained a Siamese network to match each target between two frames. Moreover, trained an online gradient classifier. However, the features used here are pixels and additional optical flow information which leads to the complex calculation; thus it is not suitable for real-time tracking. Nicolai *et al*. [27] uses the pre-trained neural network to extract the detected features and store them in the gallery. The loop iteratively compares the Euclidean distance differences between the apparent features. When matching the current detection result with the predicted trajectory, they combine the Markov distance of the Kalman filter result with the Euclidean distance of the deep feature. This improved the accuracy and also the occlusion and crossing problem. Compared to them, we use the MaskRCNN as our detector instead of ground truth for real-time tracking, we use different deep features by different loss function, and we calculate the cosine distance by using the deep feature trained with ReID dataset as the data association method. Yu *et al*. [28] uses a similar method as Nicolai*et al*., and the difference is that they trained the same neural network with Tripletloss function. He *et al*. [29] integrated the time information on the basis of Yu *et al*. [28] and achieved good results. Inspired by the above methods, we found that deep features are useful, and it is better to avoid using optical flow information to achieve real-time tracking.

3. Scale Online Deep ReID Tracker

Here we propose a Scale Online Deep ReID Tracker (SODR) framework which uses ReID metric as the appearance model to solve the MOT problem. Our hypothesis is proved in two aspects: Using ReID metric as the appearance model can solve the occlusion and crossing problem in MOT; Using the proposed scale loss can improve the ReID quality to improve the results of MOTChallenge. Figure 3 shows the whole architecture of the framework. The detectors here we use are from any public methods, and the collected $Detections$ are persons' bounding box location and appearance model(ReID metric) feature vectors that correspond to the blue bounding boxes shown in Figure 3. A deep ReID model is the center of our proposed algorithm, detailed in section 3.1. In section 3.2 we describe the Track State Estimation. Kalman filter is used here to estimate the track state. We also determine the target create and delete condition. We get the minimum cost matrix of each track by

calculating the cosine distance of current detection features and predicted track state features stored in the tracklet (a list of predicted positions by Kalman filter of this target) in the Assignment part which detailed in section 3.3. We describe the Hierarchical Association in section 3.4. Gated matrix is set to determine the effective tracklet, and the Hungarian algorithm is used here to match the detection with the track.

3.1. Deep ReID Metric

A good appearance model can make the tracker more robust, and the tracker can recognize the correct target regardless of occlusion or crossover.

In Figure 4, we illustrate our model. A Wide Residual Network [30] with a Batch Norm and ReLU layer is applied to learn deep feature representations. To learn more effective deep feature, based on softmax loss that separates features of different classes by maximizing posterior probability of the class label, we normalizes both of the weight vector of last Layer and features to remove the effect of scale variants which is formula as follows,

$$L = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i}}{\sum_{j=1}^{C} e^{\mathbf{W}_j^T \mathbf{x}_i}} \qquad (1)$$
$$s.t. \quad \| \mathbf{W}_j \| = 1, \| \mathbf{x}_i \| = 1, \forall i = 1, 2, ..., N$$

where N is the training image number, C is class number, $\mathbf{x}_i$ is the feature representations extracted from the penultimate layer of our model architecture in Figure 4, and $\mathbf{W}$ is the weight vector of the last layer of the network.

Since the norm of both weight vector $\mathbf{W}$ and feature $\mathbf{x}_i$ are normalized to be constant values, the learned feature are separable in the annular space which removes the effect of scale variations resulting in a smaller cosine angle of same-class features. The normalization constraint $\| \mathbf{x}_i \| = 1$ will make the network hard to converge explained as below. For ten class classification problem with the weight vectors $\mathbf{W}_j$, $j = 1, .., 10$, given the learned feature $\mathbf{x}_i$ of one input image, its corresponding prediction confidence is $\frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i}}{\sum_{j=1}^{10} e^{\mathbf{W}_j^T \mathbf{x}_i}}$. If we simply constraint $\mathbf{W}_j = 1$ and $\| \mathbf{x}_i \| = 1$, we have $\frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i}}{\sum_{j=1}^{10} e^{\mathbf{W}_j^T \mathbf{x}_i}} \leq \frac{e}{e+9e^{-1}} \approx 0.45$ and $-\log 0.45 = 0.346$. The loss will never converge to zero in this case. In this paper a scale parameter α is utilized to increase the fixed norm of feature vector to speed up the convergence, yet $\| \mathbf{x}_i \| = \alpha$. Then the scale loss is formulated as,

$$L_s = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{\alpha \cos(\theta_{y_i,i})}}{\sum_{j=1}^{C} e^{\alpha \cos(\theta_{j,i})}} \qquad (2)$$

where $\alpha \cos(\theta_{j,i}) = \mathbf{W}_j^T \mathbf{x}_i$ and $\theta_{j,i}$ is the angle between $\mathbf{W}_j$ and $\mathbf{x}_i$. We note that the feature separability is only determined by the angle between images which removes the effect of the scale variations of both weight vector and output feature representations.

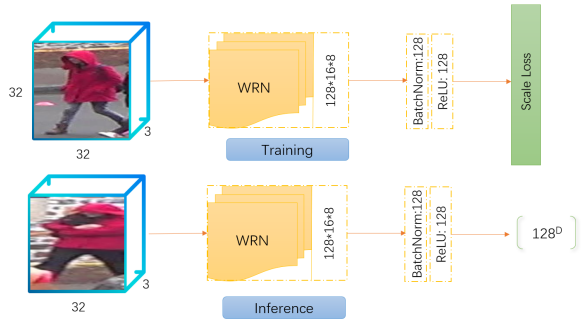

Figure 4. The WRN Network with Scale Loss architecture. The green part is our contribution, and the orange parts are others work.

With the trained ReID network, the extracted feature representation of each candidate detection at each frame is utilized to improve the occlusion and crossing problems.

3.2. Track State Estimation

Kalman filtering framework and track condition determination are followed by Bewley *et al.*'s [2] work. The description of the motion state is performed using 8 parameters, including the center coordinates of the bounding box u, v, the aspect ratio γ, the height h, and the corresponding velocity information in the image coordinate system $\bar{u}, \bar{v}, \bar{\gamma}, \bar{h}$. A standard Kalman filter based on constant velocity motion and a linear observation model is used to predict the motion state of the target and then store the prediction results in a four-dimensional matrix contains the bounding box location.

For each tracking target, the number of frames is recorded ever since the last detection result matches the tracking result. Once the detection result of a target is correctly associated with the tracking result, the parameter is set to 0. If the parameter exceeds the maximum threshold A_{max}, the tracking process for the target ends. If a target cannot always be associated with an existing tracker (the tracklet), it can be considered as a new target. If the potential new tracker can be correctly matched with the detection result in the following three frames, then it is confirmed a new moving target appears; if the requirement cannot be met, it is considered as "fake" that we need to delete the target.

3.3. Detection Assignment

Here we use the ReID features to build an appearance model to calculate the cost matrix. For each track stored in the tracklet, we calculate the minimum cosine distance of the corresponding 128^D feature vectors of the i-th track and the j-th bounding box detection d_j at the current frame,

we keep the last 50 feature vectors for each track *k*. Then the minimum cost matrix of the assignment is:

$$C_{i,j} = min\{cosine(f_j, f_k^{(i)}) | f_k^{(i)} \in \{f_k^{(i)}\}_{k=1}^{50}\} \quad (3)$$

According to the tracking methods, we maintain a gate for proper tracking. Because the motion state is estimated by Kalman filter, we need to consider the motion gate matrix. Researchers[31] have found that the Mahalanobis distance provides more possible object locations based on motion since it is scale-invariant. Here for the motion gate matrix, we calculate the Mahalanobis distance of the predicted track and the detection, and this gate matrix should be smaller than threshold $th_m = 9.4877$[2].

$$G_m = [(d_j - y_i)^T \sum_i^{-1} (d_j - y_i) \leq th_m] \quad (4)$$

Where $\sum_i^{-1}$ represents the covariance matrix between j$-th$ detection and average i-th track which means the average track location among the tracklet.

To maintain the gate matrix contain both motion and appearance information, for each row (stands for the minimum cost of a tack and the current detections) of the minimum cost matrix calculated by Equation 3, we calculate the Mahalanobis distance gate matrix and set the condition according to Equation 4.

Next, we use the Hungarian algorithm[32] to assign the detections to tracks use the minimum cost matrix as input.

3.4. Hierarchical Association

Bewley *et al*. [27] introduced a hierarchical matching strategy to solve the problem of matching priority. If a track is occluded for a long period of time, the probability of dispersion will be caused by the constant prediction of the Kalman filter. This strategy is to make more frequently seen objects have higher assigned priority. In this way, each time the tracklet with the same occlusion age A is considered. We implement our methods on this matching frame. Algorithm 1 outlines the matching algorithm:

We use the set of track indices T, corresponding to the white bounding boxes in Figure 3 and detection indices D, corresponding to the blue bounding boxes in Figure 3. The track starts at $A_{min} = 1$ and ends at $A_{max} = 20$. We compute the cost matrix and gate matrix by the cosine distance of the learned ReID features. Within the track time $n \in \{A_{min}, ..., A_{max}\}$, we do an integration to solve the assignment problem with the help of Hungarian algorithm[32]. In line 4 we assign the unmatched detection U with tracks in T_n. In line 5 and 6, we update the matched and unmatched sets, and the final matched, and the unmatched matrix is returned in line 8. To solve the sudden appearance change problem, we do an Intersection over Union (IOU)[2] check between the detection D_1 predict bounding box of track T_1 when $n = 1$ in the unconfirmed and unmatched sets. The threshold is 0.3 here for assignment.

Algorithm 1 Hierarchical Association

Input: Detection Indices $D = \{1, ..., M\}$, Track indices $T = \{1, ..., N\}$. Track start at A_{min} and end at A_{max}

1: Compute the ReID cost matrix C using equation 3 and the gate matrix G using equation 4
2: Initialize Matches *M* and Unmathes *U*
3: **for** $n \in A_{min}, ..., A_{max}$ **do**
4: $[x_{i,j}] \leftarrow matching\_results(C_{i,j}, T_n, U)$
5: $M \leftarrow M \cup \{(i,j) | G_{i,j}.x_{i,j} > 0\}$
6: $U \leftarrow U \ \{j | \sum G_{i,j}.x_{i,j} > 0\}$
7: **end for**
8: **return** M, U
9: $[x_{i,j}] = IoU\_checking$ **if** $n = 1$
10: **return** M', U'

4. Evaluation

First we explain the details of training process and the MOT Metrics we use for experiments, then we list the experiments we did: *1)* the results of two different detectors (MaskRCNN and YOLOv3), *2)* the effect of using the ReID method, *3)* the effect of different loss functions and *4)* compare our tracker with start of art tracker. We evaluate these experiments on the MOT16[5] benchmark. This benchmark evaluates tracker on seven challenge test sequences contains top-down surveillance setups and frontal-view sciences with moving cameras. The experiments with the state of art detectors are evaluated on MOT17Det benchmark. The MOT17Det uses the same video sequences as the MOT16 but with detection labels.

4.1. Datasets and Experimental Settings

Datasets. The model is trained on DukeMTMC-reID[33] ReID datasets, containing approximately 2000 pedestrians and 2000000 annotated images. We trained the model with Wide Residual Network, and designed scale loss has been used, we randomly select 100 images as a batch. We compute the feature vectors' cosine distance according to the network forward pass of each image.

Evaluation metrics. For the Multi-Person Tracking problem, we believe that an ideal evaluation index should meet the following three requirements: all the emerging targets should be found in time; find the target position to be as true as possible; logical consistency, each object should be assigned a unique track ID which stays constant throughout the sequence. These three requirements are inspired by the design of the MOT evaluation metric. To specify the results,

Metrics	Description
Rcll↑	Ratio of correctly matched detections to ground-truth detections
Prcn↑	Ratio of correctly matched detections to total results detections
MT ↑	Percentage of ground-truth trajectories which covered by the tracker output for more than 80% of their length
ML ↓	Percentage of ground-truth trajectories which covered by the tracker output for less than 20% of their length
FP ↓	Number of false positive bounding boxes
FN ↓	Number of false negative bounding boxes
IDs ↓	**Number of times that a tracked trajectory changes its matched ground-truth identity(or vice versa)**
MOTA ↑	Combines false negatives, false positives and mismatch rate
MOTP ↑	Overlap between the estimated positions and the ground truth averaged over matches

Table 1. MOT Metrics used in our experiments. Among them, the **IDs** index is particularly important to evaluate our hypothesis

Detector	Rcll ↑	Prcn ↑	GT	FP ↓	FN ↓	MODA ↑	MODP ↑
MaskRCNN	**70.7**	**78.6**	66393	12808	**19449**	51.4	79.1
YOLOv3	69.4	72.1	66393	**10052**	20298	**54.3**	**78.6**

Table 2. Results on the MOT16Challenge[5]. GT stands for Ground-Truth. We found that both MaskRCNN and YOLOv3 achieve good performance, but since our tracking method replies on the detection precision, we tend to choose the better precision detector which is MaskRCNN.

Method	Rcll ↑	Prcn ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	MOTA ↑	MOTP ↑
MaskRCNN + Correlation filter	44.2	86.3	16.8	36.4	7762	61581	929	36.3	**78.4**
MaskRCNN + Kalman filter	44.3	**86.8**	18.7	36.0	**7463**	61533	911	36.7	78.3
MaskRCNN + SODR	**49.8**	77.6	**23.6**	**25.5**	15877	**55393**	**797**	**37.4**	76.7

Table 3. Results on the MOT16Challenge[5]. We compared the traditional trackers use correlation filter and Kalman filter with the same detector MaskRCNN. Under the same condition, our method achieves good performance. In particular, the **IDs** index in our method is the lowest among the three, which proves that we solve the occlusion problem effectively.

we use ↑ denotes higher score is better, ↓ denotes that the lower score is better.

Most state-of-art algorithms choose to use the ground-truth given by them or the private detector to test the tracker's performance. However, our methods build a whole framework thus it not fair to compare with the most state-of-art trackers. For the latter experiments, we compared the results of current trackers using our framework combined with the same detector and also some state-of-art tracker benchmark used public detector such as FastRCNN[28].

4.2. Exp 1: Detectors

We test two state of art object detectors: MaskRCNN[3] and YOLOv3[34]. From the papers, we know that MaskRCNN gets an Average Precision (AP) of 37.1% and YOLOv3 is 33.0% with the same dataset COCO[35]. Test on our own computer with a GPU "GTX 1060", MaskRCNN is 5 fps while YOLOv3 is 15 fps. After adapting the data format and metric algorithm, we get the experimental results as Tabel 2 shows and the average precision curves of both detectors in Figure 5.

MaskRCNN has more precise results than YOLOv3. However, the experiments of YOLOv3 is much faster than

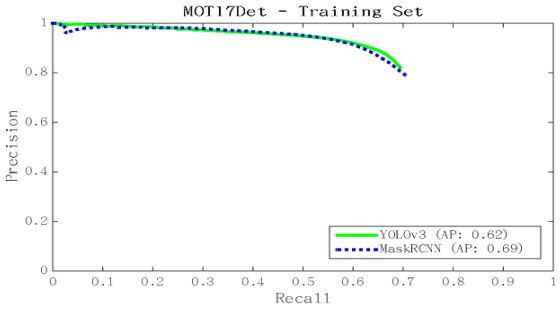

Figure 5. The average detector precision of MOTDet17. MaskRCNN achieves better accuracy in this dataset and is more stable than YOLOv3.

MaskRCNN. Since our hypothesis is solving the occlusion and crossing problem, we decide to take MaskRCNN as the detector in all later process.

4.3. Exp 2: Effect of SODR Tracker

Here we used the MaskRCNN detector combined with the traditional correlation filter Dlib tracker and also the Kalman filter tracker. The data association we used Iou checking proposed by Bewly *et al.* [2]. The experimental

Figure 6. This is the number 1 test video sequence on MOT. From top to bottom are 5 tracker:SODR (ours), cppSORT, DeepSort2, EAMTT, and GMPHD_HDA. In this expeiment, we only focus on the man with white T-shirt and black bag. There are three occlusions happen when he cross the square. **SODR:** Id_39 is our target. Result shows that there are no ID switches during the occlusions. **cppSORT:** The dark yellow bounding box is our target. There are three ID switch during the occlusions. Everytime the occlusion happen, this tracker cannot generate expected trackng results.**DeepSort2:** The orange bounding box is our target. There is one ID switch during the last occlusion. **EAMTT:** The sky blue bounding box is our target. There are two ID switches during the occlusions. **GMPHD_HDA:** The green bounding box is our target. The first occlusion cannot be tested because there is no detection result. There are two ID switch during the latter occlusions. Results show our method is the most robust one for this particular occlusion scenario. After occlusion happen, the IDs of both white T-shirt, black bag man and other people who is occluded by him are not changed.

Loss	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	MOTA ↑	MOTP↑
Softmax	21.2	33.7	**12582**	**51345**	1712	29.8	68.4
Scale_loss	**23.6**	**25.5**	15877	55393	**797**	**37.4**	**76.7**

Table 4. Results on the MOT16Challenge[5]. We compared the softmax loss and scale loss. We found that our scale loss has better accuracy than softmax loss, and notice that the false positives and false negatives are still very large, we consider this is because of the incorrect detections. The comparison between IDs index of Softmax and Scale_loss shows that better ReID model can reduce ID switches during occlusions.

results of this two trackers are compared with our proposed methods. Table 3 shows the results.

The results of the correlation filter and Kalman filter is similar, that may be because they use the same IoU[2] checking methods. In comparison to these two trackers, our methods have better "MT" and "ML" metric, that is because the Hierarchical Matching framework is suitable for not only short term but long term situations. This helps us to maintain the identities through longer occlusions. We also got lower False Negatives but higher False Positives. From the visual inspection of output shows that some pellmell response from the detector will cause false negatives. Since we consider the 20 frames as the track end condition, our "FP" is high. The "MOTA" is an important metric to test the tracker, and we got a competitive score that shows the ReID method can improve the tracking quality. Also, the "IDs" metric is lower than the other two methods which proved that when occlusion happens, our tracker is more robust. Apart from that, we found Kalman filter tracker is ten times faster than Dlib tracker, and our tracker has similar speed as the Dlib tracker.

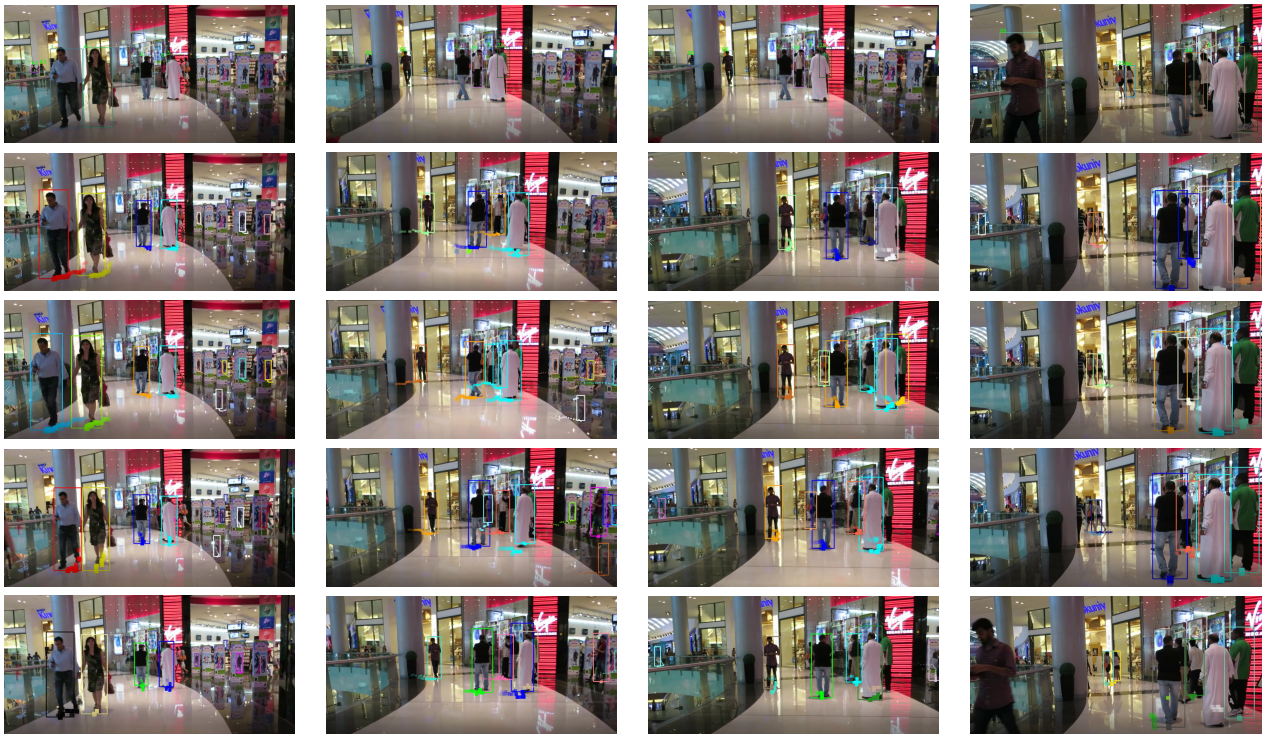

Figure 7. This is the number 12 test video sequence on MOT. From top to bottom are 5 tracker:SODR (ours), cppSORT, DeepSort2, EAMTT, and GMPHD_HDA. In this expeiment, we only focus on the man with white dress. There is a man with green T-shirt has occlusion with him. **SODR:** Id_3 is our target. Result shows that there are no ID switches during the occlusions. **cppSORT:** The light blue bounding box is our target. There is one ID switch during the occlusions. **DeepSort2:** The light blue bounding box is our target. The result is wrong for other people start from the second picture, but it still can recognize our target. Latter there is one ID switch during the occlusions. **EAMTT:** The light blue bounding box is our target. There are no ID switch during the occlusions. **GMPHD_HDA:** The blue bounding box is our target. There is one ID switch in the third picture of our target. Results show our method is the most robust one for this particular occlusion scenario. After occlusion happen, the IDs of both white dress man and green T-shirt man are not changed.

Method	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	MOTA ↑
DeepSort2(Private detector)	**32.8**	**18.2**	12852	56668	781	**61.4**
EAMTT (Public detector)	7.9	49.1	8114	102452	965	38.8
cppSORT (Public detector)	4.3	59.9	**3048**	120278	1587	31.5
GMPHD_HDA (Public detector)	4.6	59.7	5169	120970	**539**	30.5
SODR (ours- MaskRCNN detector)	23.6	25.5	15877	**55393**	797	37.4

Table 5. Results on the MOT16Challenge[5]. We compared the our tracker with state-of-art trackers. For the IDs index, Our method ranks the third position among them. This proves our hypothesis is valid and effective.

4.4. Exp 3: Evaluation of different losses

We compared two different loss functions used in our network, one is normal softmax loss, and the other one is our designed scale loss. The Table 4 shows the experimental results. The performance achieved by the designed scale loss consistently performs better than that on softmax loss, which verifies that a better ReID model can improve MOT performance.

4.5. Exp 4: Comparison with State-of-the-art

In Table 5 we compare our method to the state-of-art trackers in the same MOT dataset. However, we need to notice that most of the trackers used the ground-truth detections provided by MOTChallenge. Thus they achieved better results under the assumption that the detection output is 100% correct. Here we list some online trackers use public detections or the detections provided by themselves. We get the results directly from the MOT16Chanllenge website.

Our method is still a strong competitor to other on-

line trackers. We maintain useful "MOTA" scores, mostly tracked, mostly lost and in particular, we get the fewest false negatives.

We showed that our methods achieved a good result on MOTChallenge, also because our hypothesis focuses on occlusion and crossing problems, we cannot find a particular dataset deal with that problem. Figure 7 and Figure 6 shows two crowd scenes happen in the MOTChallenge dataset and we test with our proposed method compared with 4 State-of-Art trackers.

5. Conclusions

We presented the uses of ReID metric learning for improving the occlusion and crossing problem that traditional trackers cannot handle. We also showed that the scale loss function helps to improve the ReID quality. Our method achieved competitive results for online methods and suitable for real-time applications. Our framework may be useful for further real-time Multi-Person Tracking application.

References

[1] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3464–3468. IEEE, 2016.

[3] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[4] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

[5] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[6] Weiming Hu, Xi Li, Wenhan Luo, Xiaoqin Zhang, Stephen Maybank, and Zhongfei Zhang. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2420–2440, 2012.

[7] Jianming Zhang, Liliana Lo Presti, and Stan Sclaroff. Online multi-person tracking by tracker hierarchy. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 379–385. IEEE, 2012.

[8] Bo Yang, Chang Huang, and Ram Nevatia. Learning affinities and dependencies for multi-target tracking using a crf model. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1233–1240. IEEE, 2011.

[9] Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *European Conference on Computer Vision*, pages 605–619. Springer, 2010.

[10] Qian Yu, Gérard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[11] Bo Yang and Ram Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1918–1925. IEEE, 2012.

[12] Anton Andriyenko, Stefan Roth, and Konrad Schindler. An analytical formulation of global occlusion reasoning for multi-target tracking. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1839–1846. IEEE, 2011.

[13] Mikel Rodriguez, Josef Sivic, Ivan Laptev, and Jean-Yves Audibert. Data-driven crowd analysis in videos. In *ICCV 2011-13th International Conference on Computer Vision*, pages 1235–1242. IEEE, 2011.

[14] Donald Reid et al. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979.

[15] Dennis Mitzel and Bastian Leibe. Real-time multi-person tracking with detector assisted structure propagation. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 974–981. IEEE, 2011.

[16] Bohyung Han, Seong-Wook Joo, and Larry S Davis. Probabilistic fusion tracking using mixture kernel-based bayesian filtering. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[17] Daisuke Sugimura, Kris M Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1467–1474. IEEE, 2009.

[18] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011.

[19] Gabriel J Brostow and Roberto Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 594–601. IEEE, 2006.

[20] Mikel Rodriguez, Saad Ali, and Takeo Kanade. Tracking in unstructured crowded scenes. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1389–1396. IEEE, 2009.

[21] Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah. 2t: Multiple people multiple parts tracker. In *European Conference on Computer Vision*, pages 100–114. Springer, 2012.

[22] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1345–1352. IEEE, 2011.

[23] Brian Potetz. Efficient belief propagation for vision using linear constraint nodes. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[24] Genquan Duan, Haizhou Ai, Song Cao, and Shihong Lao. Group tracking: exploring mutual relations for multiple object tracking. In *European Conference on Computer Vision*, pages 129–143. Springer, 2012.

[25] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[26] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*, pages 215–230. Springer, 2012.

[27] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3645–3649. IEEE, 2017.

[28] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016.

[29] Qizheng He, Jianan Wu, Gang Yu, and Chi Zhang. Sot for mot. *arXiv preprint arXiv:1712.01059*, 2017.

[30] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[31] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[32] Jin Shengcan. The changed algorithm of solving assignment problem. *JOURNAL OF JIAMUSI UNIVERSITY (NATURAL SCIENCE EDITION)*, 1:029, 1998.

[33] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[34] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[35] A Augusto Alves Jr, LM Andrade Filho, AF Barbosa, I Bediaga, G Cernicchiaro, G Guerrer, HP Lima Jr, AA Machado, J Magnin, F Marujo, et al. The lhcb detector at the lhc. *Journal of instrumentation*, 3(08):S08005, 2008.

Supplement material of Multi-Person Tracking based on Person Re-identification

Hongyu Yang

I. Background introduction

With the economic and social development, surveillance video has been widely used in security, commercial, industrial production and intelligent robots and other fields. The most important object of attention in the surveillance video is persons. Understanding the behavior of persons has crucial significance for violation judgment, criminal investigation, and danger warning. How to identify, locate, and track persons is a prerequisite for understanding person behavior. Therefore, person re-identification and Multi-camera tracking are the first steps to achieve these goals.

At present, most of the work on tracking issues has focused on single-camera single-target tracking. However, this method is limited, the main difficulties are also occlusion, pose, and light. The multi-camera tracking system can well overcome these deficiencies. Therefore, the cross-camera surveillance video system is gradually gaining attention and has begun to receive in-depth research by scholars. Cross camera tracking needs to face the problem that the tracking target disappears in one camera and they may appear in other cameras. The process of retrieving a lost person target from a camera's field of view in a video taken by another camera is called person re-identification. Therefore, person recognition is the basis for cross-camera target tracking.

The most direct and typical application of person recognition is cross-camera multi-target tracking, but as an independent research topic, there are many valuable application scenarios. For cross-camera multi-target tracking, the traditional method is spatiotemporal data correlation based on some statistical methods. However, this method relies heavily on prior knowledge of statistics and can correlate targets in multiple cameras with certain probabilities, but the limitations of the effects are also obvious. Therefore, cross-camera matching based on person re-identification has now become the most mainstream solution for cross-camera tracking. .

In addition, in the current academic research, most scholars separately study person recognition and target tracking as two independent research topics. However, these two topics are actually complementary. At present, most methods of person re-identification rely on single-frame images, but the information of single-frame images is limited in the end, and cannot be solved in the case of occlusion. At present, the research of person re-recognition based on video sequences has gradually begun to attract attention. However, the cost of tracking sequence tracking is costly. The person tracking sequence can only rely on manual annotation, so a good tracking detection model is also essential.

Our topic combines these two sub-topics, trying to improve real time tracking with the help of re-identification and therefore has great research significance. This report will focus on the deep neural network based Re-identification methods and the Re-identification based tracking methods

II. person Re-identification

Person re-identification is the use of computer vision technology to determine whether there are specific persons in an image or video sequence. Widely considered as a sub-problem of image retrieval. Given a monitoring person image, retrieve the person image across devices. It is designed to compensate for the visual limitations of the current fixed camera, and can be combined with person detection/person tracking technology, and can be widely used in intelligent video surveillance, smart security, and other fields.

For cross-camera target tracking problems, when a person target disappears from one of the cameras, the person must be identified again in other cameras. This is a typical person recognition problem. In other words, person recognition technology is the basis for cross-camera tracking.

Therefore, in this section, we will first introduce existing person re-identification related data sets, accuracy assessment criteria, and some existing mainstream methods.

A. Related Data set

persons identified a total of more than a dozen related data sets. When deep learning had not yet occurred in the early years, the number of data set images at that time was still relatively small. With the advent of deep learning, person re-identification issues have greatly increased the amount of data required. This section will introduce several large-scale person identification data sets for deep learning.

(i) Market 1501
Market1501[1] is collected on the campus of Qinghua University, and the images come from six different cameras, one of which is a low-pixel camera. At the same time, the data set provides a training set and a test set. The training set contains 12,936 images and the test set contains 19,732 images. The image is automatically detected and cut by the detector and contains some detection errors (close to actual use). There were a total of 751 people in the training data and 750 in the test set. So in the training set, there is an average of 17.2 training data for each class (each person).

(ii) MARS
The MARS (Motion Analysis and Re-identification Set)[2] data set is an extension of Market1501. The image of this data set is automatically cut by the detector and contains the entire tracklet of the person image. MARS provides a total of 20,478 image sequences for 1,267 persons, and the same 6 cameras as the Market 1501. Unlike other single-frame image data sets, MARS is a large-scale person re-identification dataset that provides sequence information.

(iii) CUHK03
CUHK03[3] was collected at the Hong Kong University and the images came from 2 different cameras. This dataset provides machine automatic detection and manual detection of two data sets. The detection dataset contains some detection errors, which are closer to the actual situation. The dataset contains a total of 14,097 images of 1,467 persons, with an average of 9.6 training data per person.

(iv) CUHK-SYSU
CUHK-SYSU[4] is collected by the Hong Kong University and Zhongshan University. The feature of this data set is to provide the entire complete picture, rather than providing person images that automatically or manually extract bounding boxes, as most other data sets do. The data set contains a total of 18,184 complete images containing 99,809 person images of 8,432 persons. There are 11,206 full-length images of the training set, including 5,532 persons. The test set has 6,978 full images containing 2,900 persons.

(v) DukeMTMC-reID
DukeMTMC-reID[5] was collected at Duke University. The images were taken from 8 different cameras. The borders of the person images were manually annotated. This dataset provides training sets and test sets. The training set contains 16,522 images and the test set contains 17,661 images. There are a total of 702 people in the training data, with an average of 23.5 training data per person. The data set is currently the largest person re-identification data set and provides the annotation of personal attributes (gender/long sleeve/whether backpack, etc.).

(vi) VIPeR
The VIPeR[6] data set was an early small person re-identification data set with images from two cameras. The data set contains a total of 1,264 persons for 632 persons. Each person has two pictures taken by different cameras. The data set is randomly divided into two equal parts, one as a training set and one as a test set. Due to the earlier acquisition time, the image resolution of the data set is relatively low, so it is difficult to identify.

(vii) PRID2011

PRID2011[7] is a dataset proposed in 2011. Images are from 2 different cameras. The data set contains a total of 24,541 person images of 934 persons, so the detection frame is manually extracted. The resolution of the image size is unified of 128×64.

The above is the dataset mainly used in current person re-recognition research. As persons re-recognize pictures taken from different cameras, problems such as lighting, the person poses, viewing angles, occlusion, and image blurring may occur, causing pictures of the same person to behave differently in different cameras. Therefore, it is arduous for persons to recognize a character and it is difficult to obtain a good recognition effect by manually extracting the feature. It is necessary to learn a robust image feature through certain means.

B. Method based on Representation learning

The method based on Representation learning is a common person re-identification method[8;9;10;11]. This is mainly due to deep learning, especially the rapid development of Convolutional neural network (CNN)[12]. Since the CNN can automatically extract the representation feature from the original image data according to the task requirements, some researchers regard the person re-identification problem as the Classification/Identification problem or Verification problem. The classification problem refers to training a model using a person's ID or attribute as a training tag. The verification question refers to the input of a pair of (two) person images for the network to learn whether the two images belong to the same person.

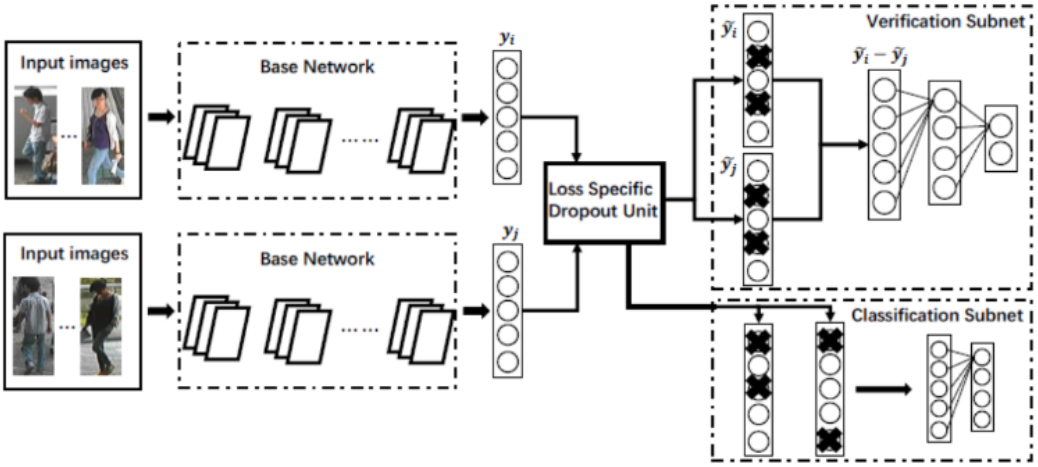

Fig. 1: Combine classification loss and verification loss[8]

Geng et al[8] uses the Classification/Identification loss and the verification loss to train the network. The network diagram is shown in Figure 1. The network input is a number of pairs of personal pictures, including Classification Subnet and Verification Sub net. The classification sub-network performs ID prediction on the picture and calculates the classification error loss based on the predicted ID. The sub-network is verified to fuse the characteristics of the two pictures and determine whether the two pictures belong to the same person. The sub-network is substantially equal to a bi-class network. After enough data training, enter a test image again and the network will automatically extract a feature that is used for person re-identification tasks.

Lin et al[9;10;11] believe that mere person ID information is not sufficient to learn a model with sufficient generalization ability. In these tasks, they additionally annotate the attributes of the person's image, such as gender, hair, and clothing.By introducing the person attribute tag, the model not only accurately predicts the person ID, but also predicts the correct person attributes, which greatly increases the generalization ability of the model. Most papers also show that this method is effective.

Figure 2 is an example. As can be seen from the figure, the characteristics of the network output are used not only to predict person ID information but also to predict various personal attributes. By combining ID loss and attribute loss, the network's generalization ability can be improved.

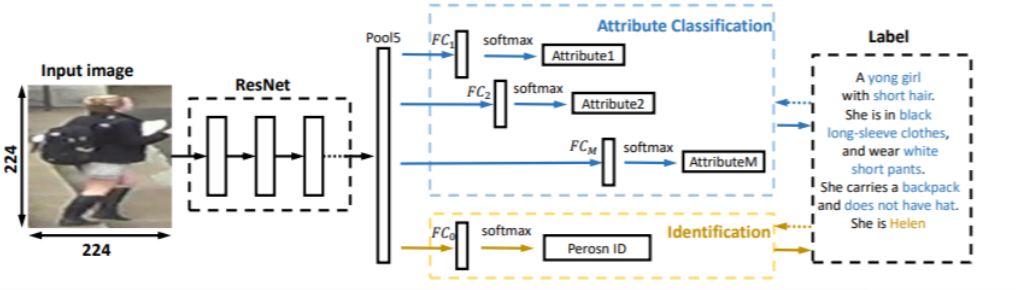

Fig. 2: Combine ID loss and attribute loss[9]

C. Method based on Metric learning

Metric learning is a method widely used for image retrieval. Different from representation learning, metric learning aims to learn the similarity of two pictures through the Internet. On the problem of person recognition, the similarity of different pictures of the same person is greater than that of different persons. Finally, the loss function of the network makes the distance of the same person image (positive sample pair) as small as possible and the distance of different person images (negative sample pairs) as large as possible. The commonly used methods for measuring learning loss are Contrastive loss[13], Triplet loss[14;15;16], Quadruplet loss[17] . First, if there are two input pictures I_1 and I_2, we can get their normalized feature vectors f_{I_1} and f_{I_2} through the network feedforward. We define the Euclidean distances of these two image feature vectors as:

$$d_{I_1,I_2} = ||f_{I_1} - f_{I_2}||_2 \tag{1}$$

(i) Contrastive loss

The contrast loss is used to train the Siamese network. The structure is shown in Figure 3. The input of the twins network is a pair of (two) pictures, I_a and I_b, which can be the same person or different persons. Each pair of training pictures has a tag y, where $y = 1$ means that the two pictures belong to the same person (positive sample pairs), whereas $y = 0$ means they belong to different persons (negative sample pairs). Contrast loss function writing:

$$L_c = yd^2_{I_a,I_b} + (1-y)(\alpha - d_{I_a,I_b})^2_+ \tag{2}$$

Where $(z)_+$ represents $max(z, 0)$, α is a threshold parameter designed based on actual needs. In order to minimize the loss function, when the network inputs a pair of positive samples, $d(I_a, I_b)$ will gradually become smaller, that is, person images with the same ID will gradually form clusters in the feature space. Conversely, when the network inputs a pair of negative samples, $d(I_a, I_b)$ gradually increases until it exceeds the set α. By minimizing L_c, the distance between pairs of positive samples can be gradually reduced, and the distance between pairs of negative samples gradually becomes larger to meet the needs of persons to re-identify tasks.

(ii) Triplet loss

Triplet loss is a widely used measure of learning loss, followed by a large number of metric learning methods based on the evolution of triple loss. As the name implies, the triple loss requires three input pictures. Unlike contrast loss, an input Triplet includes a pair of positive sample pairs and a pair of negative sample pairs. The three images are named Anchor a, Positive p, and Negative n.

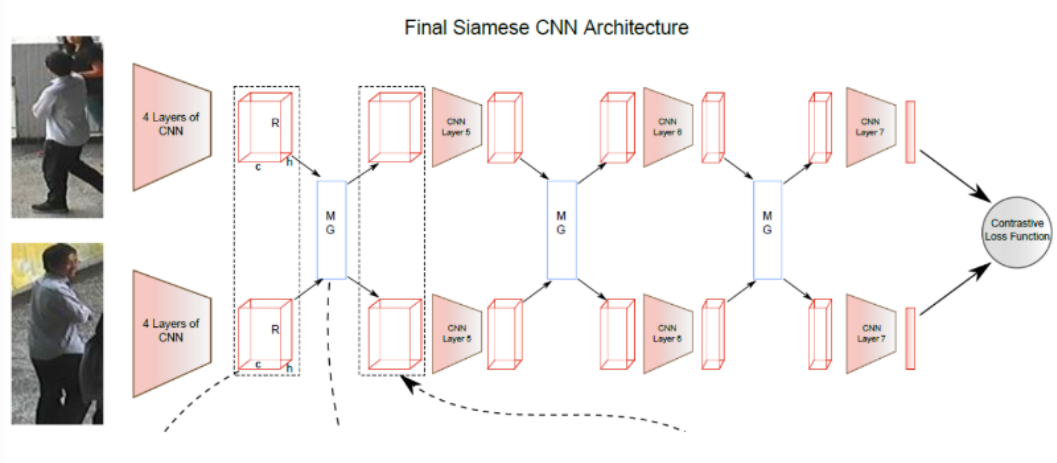

Fig. 3: Siaseme architecture[13]

The image a and the image p are a pair of positive samples, and the image a and the image n are a pair of negative pairs. The triple loss is expressed as:

$$L_t = (d_{a,p} - d_{a,n} + \alpha)_+ \tag{3}$$

As shown in FIG. 4, the triple can pull the distance between the positive sample pairs, push away the distance between the negative sample pairs, and finally make the person images with the same ID form a cluster in the feature space, achieving person recognition.

Fig. 4: Triplet loss[18]

Cheng et al[16] think that the formula (3) only considers the relative distance between positive and negative sample pairs, and does not consider the absolute distance between positive sample pairs. For this reason, improved triplet loss is proposed:

$$L_i t = d_{a,p} + (d_{a,p} - d_{a,n} + \alpha)_+ \tag{4}$$

The formula (4) adds $d_{a,p}$ to ensure that the network not only pushes the positive and negative samples in the feature space, but also ensures that the positive sample pairs are in close proximity.

(iii) Quadruplet loss

A quadruple loss is another improved version of the triple loss. The quadruplet requires four input pictures, which have a negative sample picture. That is, the four pictures are a fixed picture a, a positive sample p, a negative sample picture 1(Negative1) $n1$ and a negative sample picture 2(Negative2) $n2$. Among them, $n1$ and $n2$ are pictures of two different person IDs. The structure is shown in Figure 5 and the quadruplet loss is expressed as:

$$L_q = (d_{a,p} - d_{a,n1} + \alpha)_+ + (d_{a,p} - d_{n1,n2} + \beta)_+ \tag{5}$$

Where α and β are manually set constants, usually β is set to less than α, the former is called strong promotion, and the latter is called weak promotion. Compared to the triple loss, which only

considers the relative distance between the positive and negative samples, the second term added to the quad does not share the ID, so the absolute distance between positive and negative samples is considered. Therefore, the quad loss usually allows the model to learn better features.

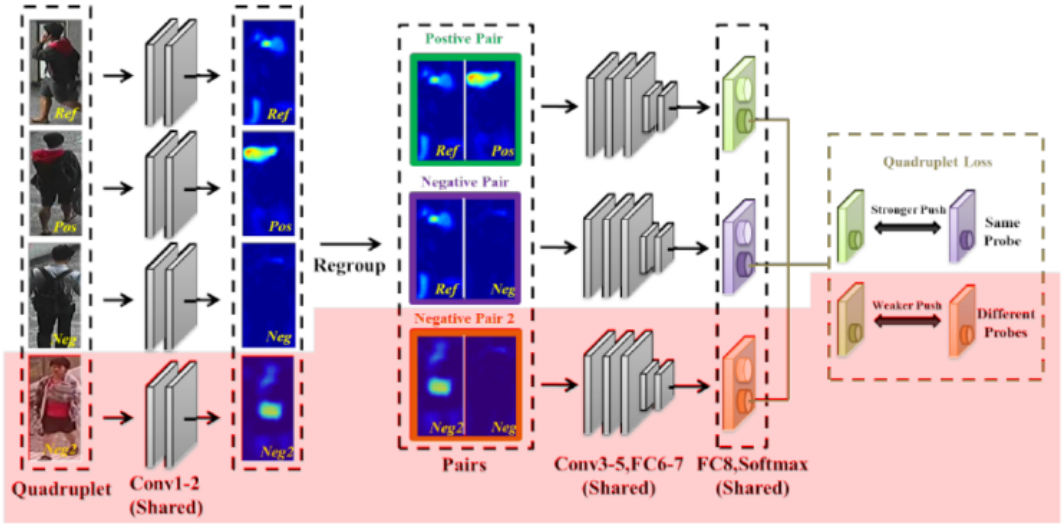

Fig. 5: Quadruplet loss architecture[17]

(iv) Triplet loss with hard sample mining

The hard-sample triad loss (TriHard) is an improved version of the triad loss. Traditional triads randomly sample three images from training data. Although this approach is relatively simple, most of the sampled pictures are simple and easily distinguishable pairs. If a large number of trained sample pairs are simple sample pairs, then this is not conducive to better characterization of network learning. A large number of papers have found that using harder samples to train the network can improve the generalization ability of the network. One paper[19] proposed an online hard sample sampling method based on training batches - TriHard. The core idea of TriHard is that for each training batch, randomly select P ID persons, and each person randomly selects K different pictures, that is, a batch contains $P \times K$ pictures. Afterward for each picture in the batch a, we can pick one of the hardest positive samples and one of the hardest negative samples and a to form a triple. First of all, we define an image set with a as the same ID as A, and a set of image images with different IDs as B, then TriHard denotes:

$$L_{th} = \frac{1}{P \times K} \sum_{a \in batch} (\max_{p \in A} d_{a,p} - \min_{n \in B} d_{a,n} + \alpha)_+ \tag{6}$$

Where α is the artificially set threshold parameter. The TriHard loss calculates the Euclidean distance of each image in the a and the batch in the feature space, then selects the positive sample p that is the farthest (much less) than the a distance and the closest (most like) distance. The negative sample n is used to calculate the triple loss. TriHard loss is usually better than the traditional triple loss.

D. Method based on Local feature

The classification of the network's training loss function can be divided into representation learning and metric learning. The related methods have been introduced previously. From the aspect of extracting image features, the method of person recognition can be divided into a global feature and a local feature-based method. The global feature means that the network extracts a feature from the entire image. This feature does not consider some local information. Local features refer to manually or automatically letting the network focus on key local areas and then extract the local features of these areas. The commonly used

methods for extracting local features include image segmentation, positioning using skeleton key points, and attitude correction.

Image dicing is a common way to extract local features[20;21]. As shown in 6, the picture is vertically divided into several pieces, because vertical cutting is more in line with our intuitive perception of human recognition, so horizontal recognition is rarely used in person recognition. Afterward, the segmented image blocks are sent to a long short-term memory network (LSTM), and the last feature merges the local features of all the image blocks. However, the disadvantage is that the requirement for image alignment is relatively high. If the two images are not aligned up and down, then the head and upper body are likely to be compared. This makes the model wrong.

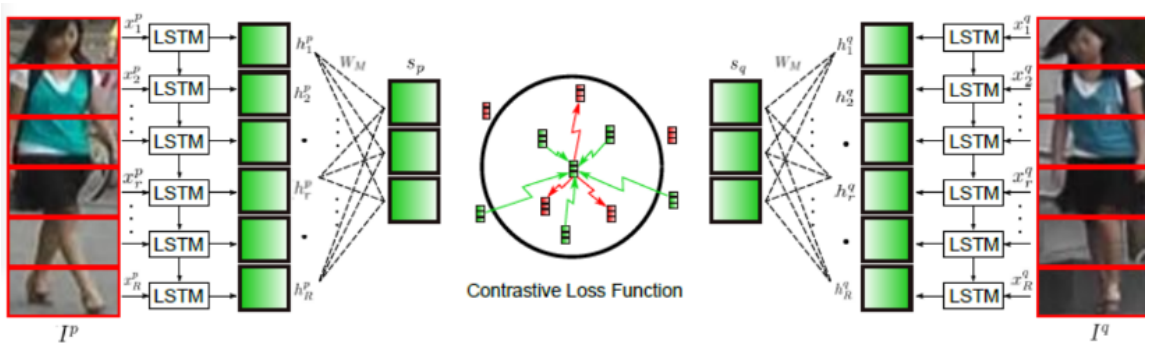

Fig. 6: Using image dice to extract local feature[21]

In order to solve the problem of manual image slice failure under image misalignment, some papers use some prior knowledge to align Persons. These prior knowledge are mainly pre-trained Pose and Skeleton models.

The paper[22] first uses the model of pose estimation to estimate the key points of the person and then uses the affine transformation to align the same key points. As shown in 7, a person is usually divided into 14 key points that divide the body's results into several regions. In order to extract local features at different scales, the author has set three different PoseBox combinations. The three PoseBox corrected pictures are sent to the network along with the original corrected pictures to extract features. This feature contains global information and local information. In particular, it is proposed that if this affine transformation can be performed in the pre-processing before entering the network, it can also be performed after input into the network. If it is the latter then it needs to make an improvement on the affine transformation, because the traditional radiation change is not guidable. In order for the network to be trained, it is necessary to introduce a derivative that approximates radiation changes.

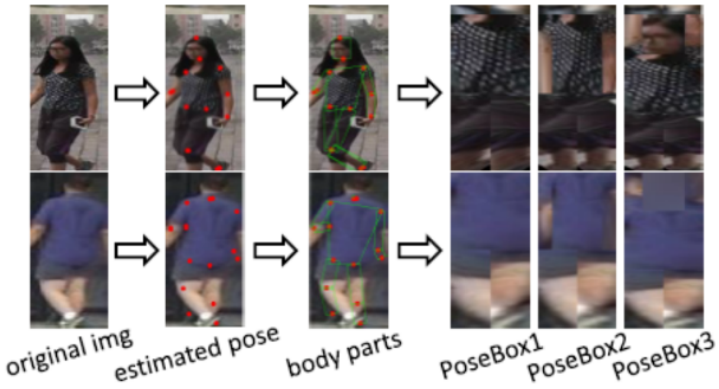

Fig. 7: Pose box architecture[22]

CVPR2017's work Spindle Net[23] also uses 14 human body key points to extract local features. Unlike paper[22], Spindle Net does not use affine transformations to align local image regions. Instead, it uses these key points directly to derive the Region of Interest (ROI). The Spindle Net network is shown as

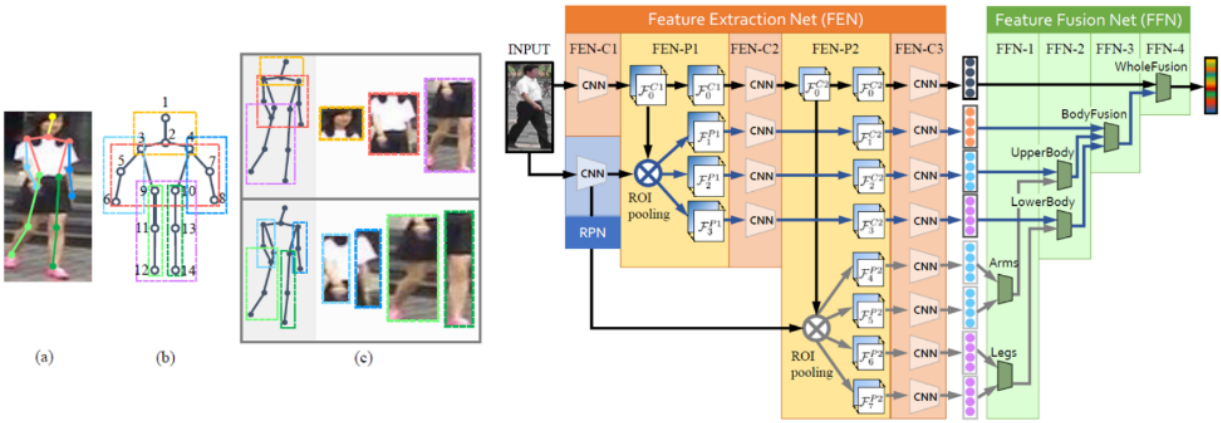

Fig. 8: Spindle Net architecture[23]

8. First, 14 key points are extracted from the skeleton key points, and then 7 ROIs are extracted using these key points. The CNN (in orange) parameters of all extracted features in the network are shared. This CNN is divided into three linear sub-networks FEN-C1, FEN-C2, and FEN-C3.

For the input of a person image, a pre-trained skeleton key extract CNN (blue) to obtain 14 human body key points, resulting in 7 ROI regions, including three large regions (head, upper body, Lower body) and a small area of four limbs. The seven ROI regions and the original picture enter the same CNN network to extract features. The original image gets a global feature through the full CNN. Three large areas get three local features through the FEN-C2 and FEN-C3 subnetworks. The four limb regions receive four local features through the FEN-C3 subnetwork. Afterward, these eight features are connected at different scales according to the illustrated method, and finally, a person re-identification feature that combines global features and multiple-scale local features is obtained.

The paper[24] proposed a Global-Local-Alignment Descriptor (GLAD) to solve the person pose change problem. Similar to Spindle Net, GLAD uses the extracted key points of the body to divide the picture into three parts: head, upper body, and lower body. After that, the whole image and the three partial images are input together into a parameter-sharing CNN network. Finally, the extracted features integrate global and local features.

In order to adapt to the input of pictures of different resolution sizes, the network uses global average pooling (GAP) to extract the respective features. Slightly different from Spindle Net is that the four input pictures each calculate the corresponding loss instead of merging a total loss for a feature.

All of the above local feature alignment methods require an additional skeleton keypoint or pose estimation model. Training a model that can reach a practical level requires the collection of enough training data. The cost is high. In order to solve the above problem, AlignedReID[25] proposes an auto-alignment model based on SP distance, which automatically aligns local features without additional information. The method used is dynamic alignment algorithm, or also called as shortest path distance. The shortest distance is automatically calculated.

The core idea of this paper is that they calculate the shortest distance between two local features to align different parts, and then they only keep the global feature to measure the difference between pictures. The distance of the person in different pictures can be summed by the global distance and focal distance. Global distance is defined by *L2* distance while the local distance is calculated by dynamic aligning.

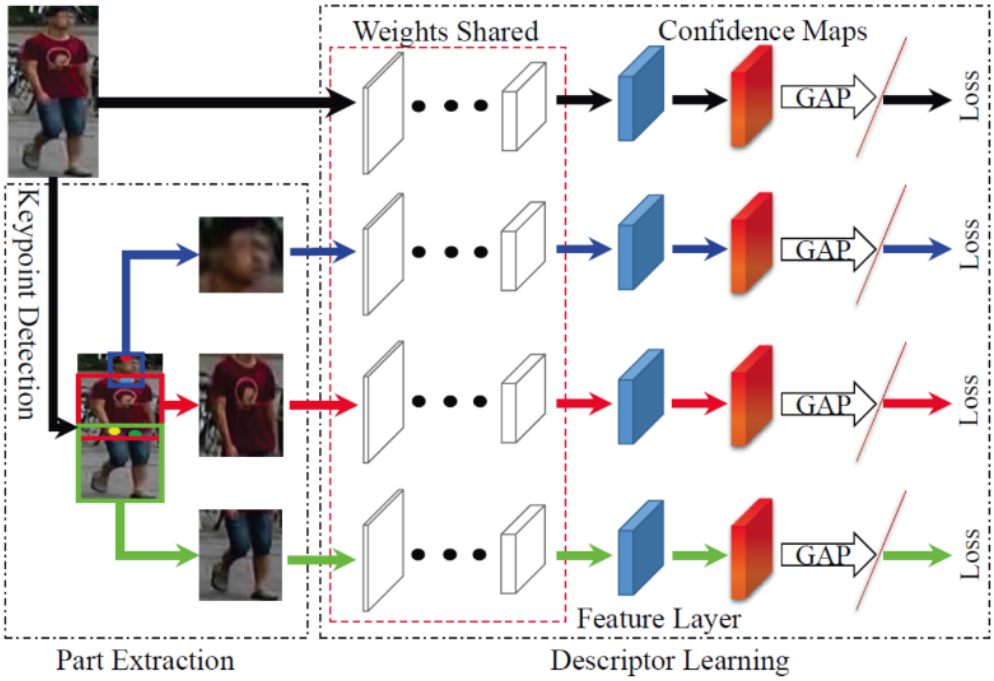

Fig. 9: GLAD architecture[24]

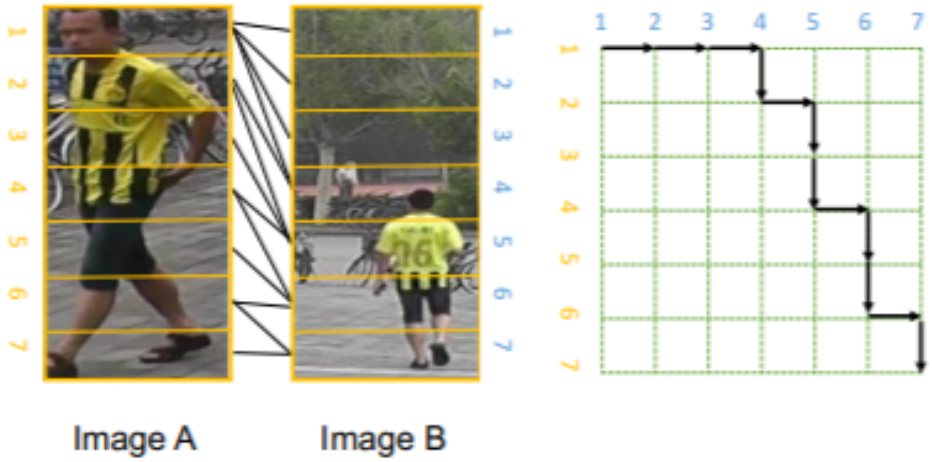

Fig. 10: Shortest distance calculated by the local feature[25]

Given the local feature of two pictures, $F = f_1, f_2, ...f_h$, $g = g_1, g_2, ...g_h$. First they use element-wise to regularize the distance between 0 and 1:

$$d_{i,j} = \frac{e^{\|f_i - g_j\|_2 - 1}}{e^{\|f_i - g_j\|_2 + 1}}, \qquad i, j \in 1, 2, 3, ...H \tag{7}$$

Where $d_{i,j}$ represents the distance of the ith vertical bar in the first picture and the jth vertical bar in the second picture. Then a distance martix $\mathbf{D}$ made up of these distances. They define the shortest distance as from $(1, 1)$ to (H, H) in the distance matrix. This can be calculated by the dynamic programming:

$$S_{i,j} = \begin{cases} d_{i,j} & i = 1, j = 1 \\ S_{i-1,j} + d_{i,j} & i \neq 1, j = 1 \\ S_{i,j-1} + d_{i,j} & i = 1, j \neq 1 \\ min(S_{i-1,j}, S_{i,j-1}) + d_{i,j} & i \neq 1, j \neq 1 \end{cases} \tag{8}$$

The local distance can be combined with any other global distances, then they chose TriHard loss as their metric learning baseline. The whole framework as below:

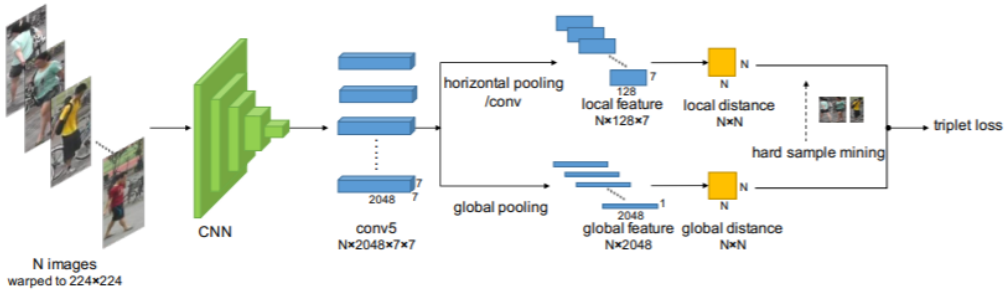

Fig. 11: Aligned SP architecture[25]

E. Method based on Video sequence

The above methods are all based on the single-frame image method. Usually, the information of a single frame of image is limited. Therefore, there is a lot of work focused on the use of video sequences for person re-recognition methods[26;27;28;29;30;31;32]. The main difference in the method based on video sequences is that this kind of method not only considers the content information of the image but also considers the motion information between frames and frames.

The main idea of the method based on single-frame images is to use CNN to extract the spatial features of images. The main idea of video sequence-based methods is to use CNN to extract spatial features while using Recurrent Neural Networks (RNN) to extract temporal features. The diagram 12 is a typical idea. The network input is a sequence of images. Each image is extracted through a shared CNN to extract image space content features. These feature vectors are then inputted to an RNN network to extract the final features. The final feature combines the content features of single-frame images and the motion characteristics between frames. This feature is used to train the network in place of the image features of the previous single-frame method.

One of the representative methods of the video sequence class is Accumulated Motion Context Network (AMOC)[32]. The AMOC input includes the original image sequence and the extracted optical stream sequence. The extraction of optical flow information usually requires the use of traditional optical flow extraction algorithms, but these algorithms are computationally time to consume and are not compatible with deep learning networks. In order to get a network that automatically extracts optical streams, the author first trained a Motion Information Network (Moti Nets). This motion network input is the original image sequence, and the label is the optical flow sequence extracted by the conventional method. As shown in 13, the original image sequence is displayed in the first row, and the extracted optical stream sequence is displayed in the second row. The network has three optical flow prediction outputs, namely Pred1, Pred2, and Pred3. These three outputs can predict optical flow diagrams at three different scales. Finally, the network integrates the optical flow prediction output at three scales to obtain the final optical flow diagram. The predicted optical flow sequence is shown in the third row. By minimizing the errors of the predicted optical flow diagram and the extracted optical flow diagram, the network can extract more accurate motion characteristics.

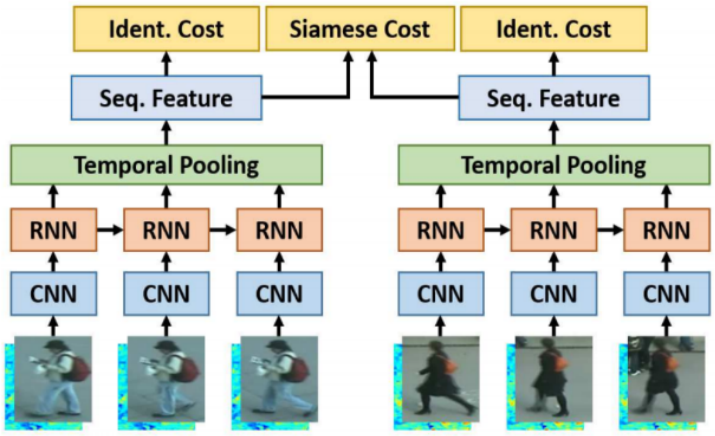

Fig. 12: person re-recognition network structure diagram based on video sequence[30]

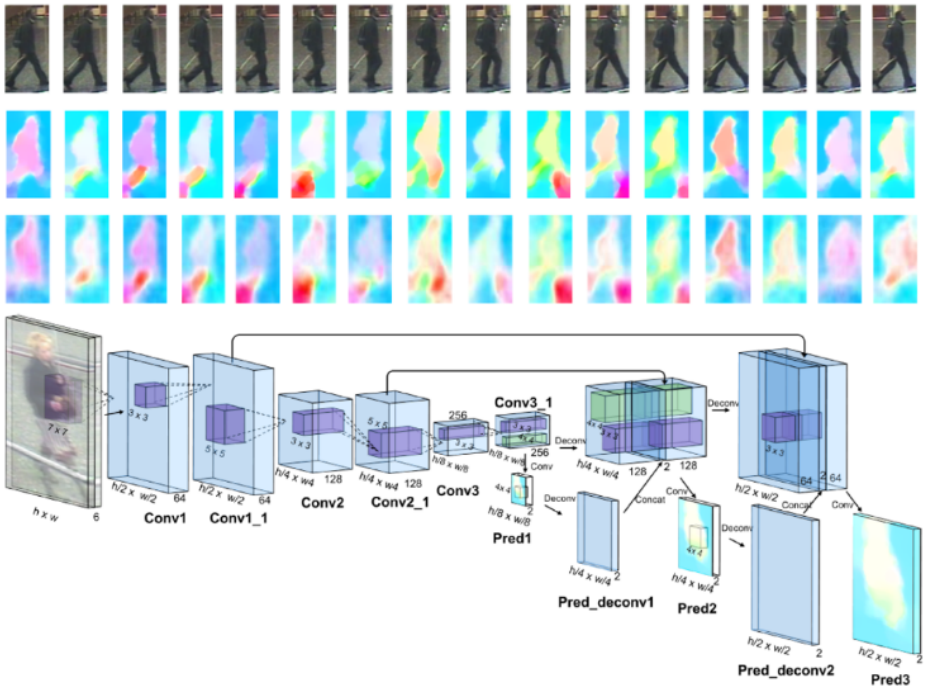

Fig. 13: Schematic diagram of motion network[31]

The core idea of AMOC is that in addition to extracting the features of the sequence image, the network also needs to extract the motion features of the moving optical stream. The network structure diagram is shown in Figure 14. AMOC has two sub-networks of Spatial Network, Spat Nets, and Sports Information Network. Each frame of the image sequence is input to the Spat Nets to extract the image's global content features. The two adjacent frames will be sent to Moti Nets to extract the characteristics of the optical flow diagram. Then the spatial features and optical flow features are merged and input into an RNN to extract temporal features. Through the AMOC network, each image sequence can be extracted to incorporate features of content information and motion information. The network uses classification loss and contrast loss to train the model. The feature of sequence image combined with motion information can improve the accuracy of person recognition.

Paper[32] shows from another point of view the effect of multi-frame sequences to compensate for the

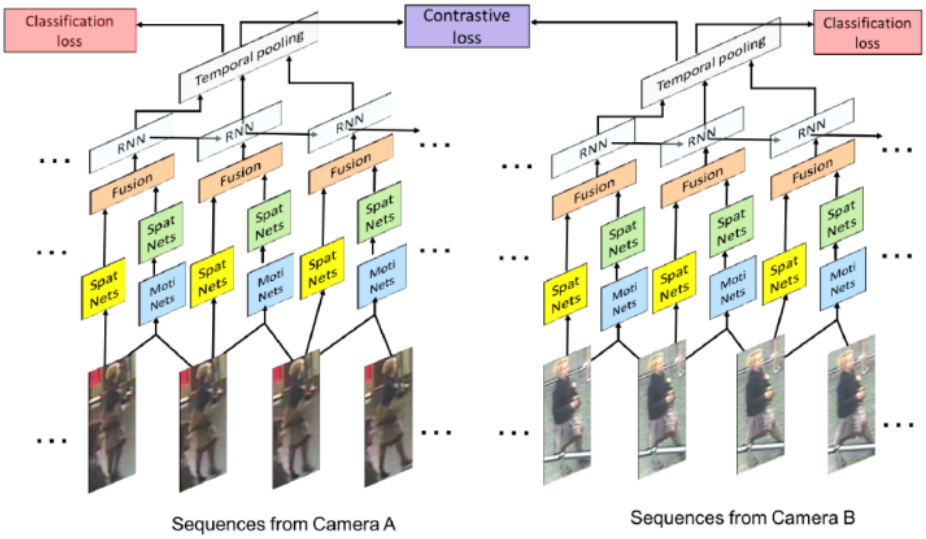

Fig. 14: AMOC structure diagram[32]

insufficiency of single-frame information. At present, most video-based ReID methods still don't care whether the sequence information is lost to the network or not, allowing the network to learn its own usefulness. Information does not intuitively explain why multi-frame information is useful. Song et al clearly pointed out that when a single frame image encounters occlusion and other conditions, it can be compensated by other information of multiple frames, directly inducing the network to perform a quality judgment on the picture, and reducing the quality of the frame with poor quality.

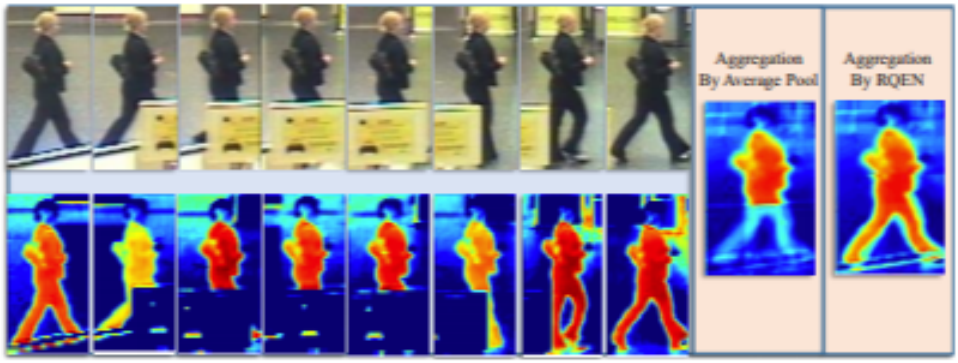

Fig. 15: Frame quality judgment[32]

As shown in the below figure, the article considers that if the masking is more serious, if the ordinary pooling will cause the deterioration of the intention map, the characteristics of the occlusion area will be lost. Using the paper's method to make a quality judgment every frame, we can focus on those relatively complete frames, making the attention map more complete. The key implementation is to use a network of pose estimation. The paper is called a landmark detector. When the coverage is incomplete, it is proved that there is occlusion, and the picture quality will be degraded. Afterward, both the pose feature map and the global feature map are input to the network at the same time, allowing the network to make a weighty decision for each frame, placing high-quality frames with high weights, and then performing a linear superposition of the feature map.

Fig. 16: Generated pictures by GAN[34]

F. Method based on GAN

One big problem with ReID is that it is difficult to obtain data. Until the CVPR18 deadline is finalized, the largest ReID dataset is a few thousand IDs and tens of thousands of pictures (the sequence is assumed to be only one). So after ICCV17 GAN was applied to ReID, a lot of work on GAN emerged.

Zheng et al's paper[34] was the first one to use GAN as a ReID and was published at ICCV17. As shown below, the quality of the image generated by this paper is not soaring. Another problem is that since the image is generated randomly, it means that no label can be used. In order to solve this problem, the paper proposes a label smoothing method that is to take the value of each element of the label vector to be the same, and to satisfy the sum of 1. The generated image is added to the training as training data. Since the baseline at the time is not as high as it is now, the effect is quite obvious. At least a large amount of data can effectively avoid overfitting.

Zhong et al[35] improved the above method. The previous GAN mapping was still random but became a controllable generator in this article. One problem with ReID is that there are biases in different cameras. This bias may come from various factors such as light and angle. To overcome this problem, the paper uses GAN to transfer a camera's picture to another camera. A smoothing parameter was added to the paper. Experiments have shown that this works well. The final overall network framework is as follows:

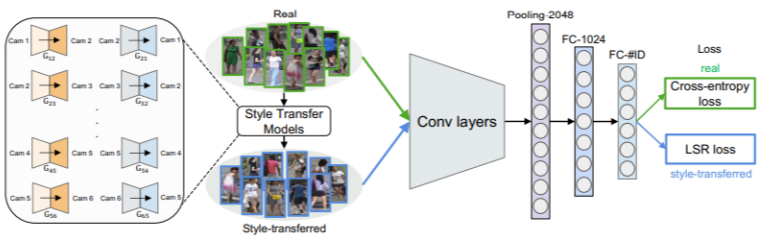

Fig. 17: Controlled GAN archtecture[35]

In addition to the bias of the camera, there is a problem with the ReID dataset bias, a large part of this bias is caused by the environment. To overcome this bias, Wei et al[36] uses GAN to transfer persons from one data set to another. In order to achieve this migration, GAN's loss is slightly designed, one is

the absolute error of the foreground loss, and the other is a normal discriminator loss. The determiner loss is used to determine which domain the generated graph belongs to, and the loss of the foreground is to ensure that the prospects of persons are as realistic as possible. This foreground mask is obtained using PSPnet, as shown below. Another contribution of the paper is to propose an MSMT17 data set, but it has not yet been published.

Another difficulty in ReID is the difference in posture. To overcome this problem, the paper[37] uses GAN to create a series of standard pose images. The paper has extracted a total of eight poses. The eight poses basically cover all angles. Every picture generates such a standard 8 pose, so different pose problems are solved. Finally, use the features of these images to perform an average pooling to get the final feature. This feature combines the information of various poses and solves the pose bias problem. This job made a single query into a multi-query. This work also requires a pre-trained pose estimation network for pose extraction.

III. Multi-camera Tracking

ReID is an image retrieval problem. It extracts a feature from the detected person image and judges the similarity of the two images according to the feature to achieve the purpose of retrieval. Tracking is more like a data association problem. Using ReID features, spatiotemporal information, and motion information, etc., to associate two objects to a match. One of the tracking methods is tracking by detecting, and ReID does this kind of tracking. The main idea is to first detect the Person target, and then determine if the bounding box of the defect belongs to the same Person, and associate the detection box of the same Person with the tracklet.

A. Related Data set

(i) DukeMTMC

DukeMTMC[38] is a cross-camera multi-target tracking data set that has been build by five doctoral students at Duke University for more than one year. It is currently the best and latest MTMC data set.

The dataset includes a total of 85 fixed-camera 85-minute video data, and the video is a 60-fps 1080p resolution image. A total of 2,000,000 frames of image data were manually annotated, of

Fig. 18: Generated pictures by controlled GAN[36]

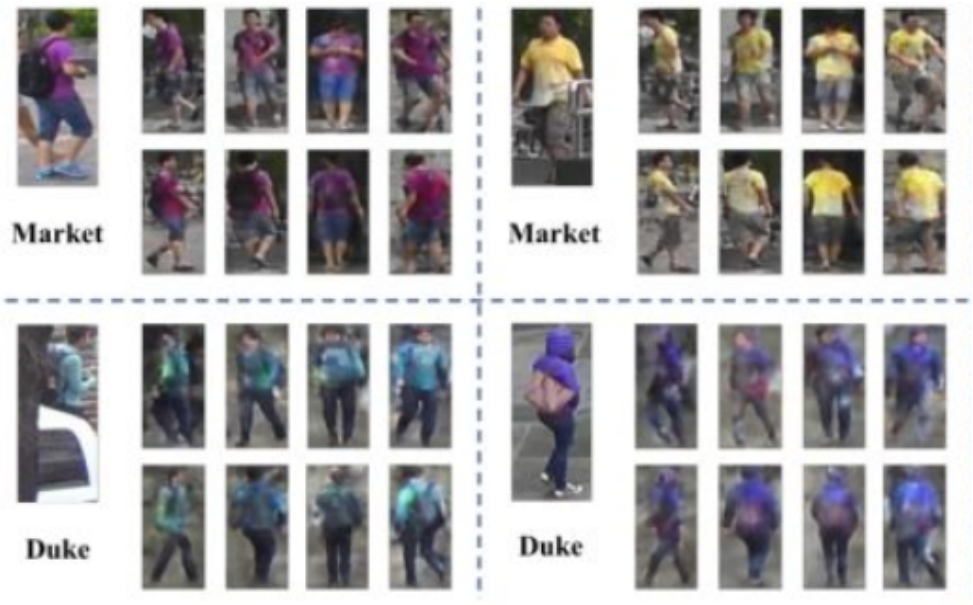

Fig. 19: Pose unbias GAN[37]

which more than 2,000 were Persons, more than all the current MTMC data sets. All the tracking sequences add up to more than 30 hours. For a single camera, the single-frame image contains a minimum of 0 people and a maximum of 54 people. There were a total of 4,159 trajectory shifts and 50 trajectory blind spots, in addition to 1,800 self-occlusion. There is a slight overlap in the field of view of the two pairs of cameras, so this can be used to study coincident cross-camera tracking as well as non-coincident cross-camera tracking. The video for the first 5 minutes of each camera is used as a training set or verification set, and the remaining 80 minutes are used as a test set. A total of 891 persons appeared in only one camera. The tracker easily generates FP, which is a massive test for the tracker.

In addition to annotating the ID and tracking sequence of the person, the DukeMTMC dataset also provides camera calibration data for 8 cameras, which provides both cameras internal and camera external data, taking into account the temporal and spatial information related issues. These camera calibration data allow researchers to obtain the space-time coordinates of the tracking target in the world coordinate system. This information is favorable for filter-related methods.

(ii) MOT16

MOT16[39] is a single-camera multi-target tracking data set that contains a total of 14 video data, of which 7 training data and the remaining 7 are test data.

These videos come from 7 different scenes. Each scene's video is randomly cut into two parts that do not coincide and are used as the training and test data. Unlike Duke MTMC, Duke MTMC only focuses on personal goals, while the MOT dataset focuses on personal goals, followed by 12 common targets such as cars, bicycles, and motorcycles. Because the video sources are different from each other, the video's resolution and frame rate are different. The video is shot from both a fixed and handheld position. Overall, it is a diverse and challenging multi-goal. Tracking data sets. The MOT16 data set has very precise labeling. Each target's detection frame is finely aligned. There is almost no pixel in the leaked frame target and no extra pixels are consumed. That is, the frame boundary of the target is basically defined. Finally, the MOT16 data set has a total of 215,166 detection frames, with an average of 19.15 frames per tracking target. Most of them are ordinary persons.

(iii) PETS16

PETS16[40] is a multi-target tracking data set that contains a total of 14 video data, of which 6 are training sets and 8 are test sets. The video is divided into four resolutions: 480p, 512p, 960p, and 1280p, and the frame rate is 25fps or 30fps. A total of 7,051 test frames were marked in the training set, and 8,025 test frames were marked in the test set.
The dataset mainly provides tracking sequence markings for persons walking on the road and ships moving on the water surface. The shooting angles are basically low-altitude cameras, such as those taken from the driver's seat of the truck.

B. Method based on Appearance model

The Appearance model includes both the visual features of the target and the similarity and dissimilarity measures between the targets. Visual expression is certainly based on image features. Before deep learning methods appear, scholars often manually extract some traditional features. Because the topic focuses on deep learning features, only brief introductions to these traditional methods are introduced in this report. Traditional image features include:

- Point feature, such as Harris corner, SIFT corner, SURF corner, etc.
- Color/intensity features, such as the simplest templates, color histograms, etc.
- Optical flow, containing time domain information
- Gradient/pixel-comparison features, typical of HOG features
- Region covariance matrix features, this feature is relatively robust to lighting and scale transformations
- Depth, the depth information, is still quite large for video 3D data

These image features have a wide range of applications in traditional tracking, but with the development of deep learning and person re-identification, ReID features have gradually become an excellent appearance model[40;41].

Beyer et al[41] combine the ReID feature with some other data associations (DA), given the two frames as I_1 and I_2. Their distances (in inverse proportion to their similarities) are expressed as follows: :

$$d(I_1, I_2) = \frac{d_{pos}(I_1, I_2)}{N_{pos}} \frac{d_{app}(I_1, I_2)}{N_{app}} \tag{9}$$

Where d_{app} is the distance of the ReID eature of the two pictures, such as the most commonly used Euclidean distance. N_{pos} and N_{app} are normalized parameters, allowing d_{pos} and d_{app} to be in the same order of magnitude. d_{pos} is a traditional data association method cccccccc Leal-Taixe et al's work[41] is even more straightforward. It is necessary to match the two detection boxes by training a twin network. As shown in 20, the detector detects a number of detections, and then associates the same tracklet with a trained person recognition network, and then passes through a linear programming. The method gets the final Trajectory.

This method is relatively simple and requires only one detector and a ReID model to achieve the multi-camera tracking problem, which is a common method in the industry. However, the disadvantages of this method are also obvious. It depends much on the performance of the detector and the ReID model.

In our topic, we do not pay attention to the research of the detector, that is, work under the premise of having a good detector. The final performance of this method is completely affected by the ReID model.

C. Method based on Correlation filtering

Correlation filtering is a common object tracking method. Bayesian filter, Kalman filter, particle filter and so on are all applied to this problem. However, on cross-camera target tracking issues, there is not much-related work and there is even less work to combine with person recognition. Beyer et al[41] combined Bayesian filter with ReID and achieved the goal of multi-camera tracking.

Given a ReID model f_θ, the image I_p can get an embedded feature $e_p = f_\theta(I_p)$. For a complete frame image I, you can get a $D_I(e_p) = (||e_{i,j} - e_p||)_{i,j}$, $e_{i,j}$ represents the distance between the image slice

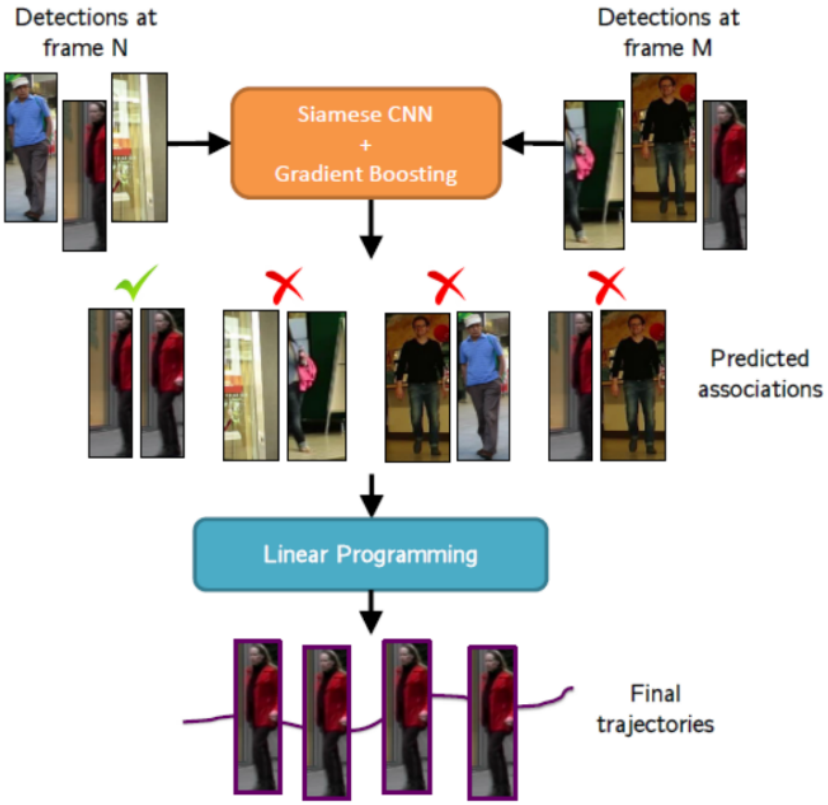

Fig. 20: Multi-camera tracking based on person re-identification of twin networks[42]

center and the ReID feature of the target image. The smaller the distance is, the more similar they are. Finally, an embedding distance map D_I can be obtained. This D_I can then be converted into a Bayesian filter observation. The observation model is expressed as:

$$P(z_t|X_t, z_{1:t-1}) = softmin(D_I(f(Xt, z_{1:t-1}))) \tag{10}$$

Among them, $e_p = f(Xt, z_{1:t-1})$ can be updated in many ways. In the paper, it simply replaces the first occurrence and does not update, ie $e_p = f_\theta(z_1)$. Of course, in order to adapt to this new observer model, we need to reconstruct the traditional optimal Bayesian filter and reconstruct Bayesian rules using probability. The final expression is as follows:

$$P(X_t|z_t, z_{1:t-1}) \propto \overbrace{P(z_t|X_t, z_{1:t-1})}^{\text{new measurement}} \overbrace{P(X_t|z_{1:t-1})}^{\text{belief propagation}} \tag{11}$$

The publicity is divided into two parts. The first part is the latest observations, and the latter part can be used to estimate the state from the previous moment. When we have the observation model, the next step is to use a Bayesian filter for state estimation. The latter item of the formula (11) can be further decomposed using the full probability model and Markov rules:

$$P(X_t|z_{1:t-1}) = \int \overbrace{P(X_t|x_{t-1})}^{\text{dynamics model}} P(x_{t-1}|Z_{1:t-1}) dx_{t-1} \tag{12}$$

This formula can use the state X_{t-1} at the previous moment and the observation $z_{1:t-1}$ at all times to estimate the current state X_t. $P(X_t|x_{t-1})$ represents a dynamic model, which compares typical person

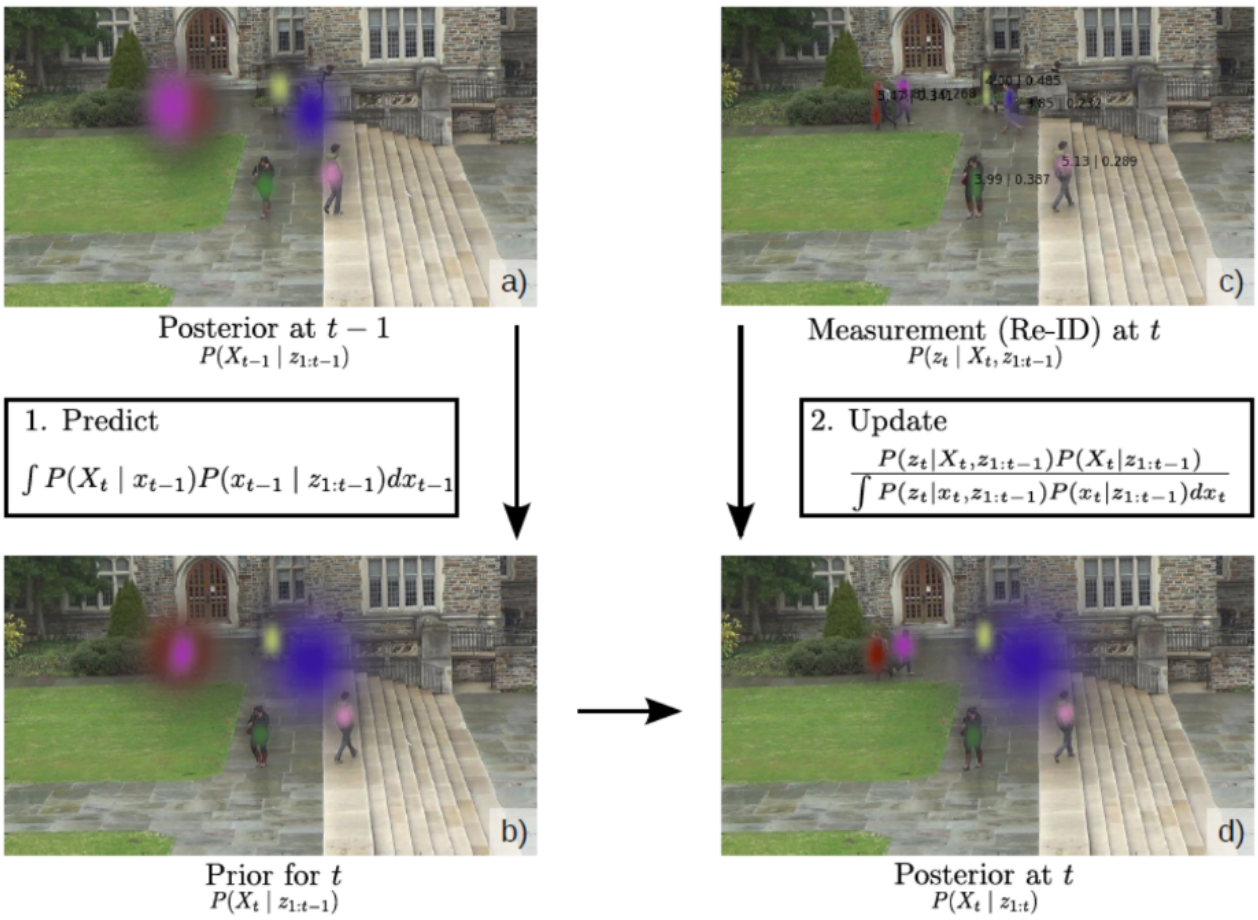

Fig. 21: Multi-target tracking based on person re-identification and Bayesian filtering[41]

constant speed constraints in a motion model such as in multi-camera tracking. With a new observation z_t input, the posterior probability can be updated according to the Bayesian rule:

$$P(X_t|z_{1:t}) = \frac{\overbrace{P(z_t|X_t, z_{1:t-1})}^{\text{measurement model}} P(X_t|z_{1:t-1})}{\int P(z_t|x_t, z_{1:t-1}) P(x_t|z_{1:t-1}) d_{x_t}} \tag{13}$$

This is related to the formula (10). Specifically, when there is no observation result, $P(zt|X_t, z_{1:t-1})$ is uniformly distributed, so it can be cancelled after normalization. . That is, the posterior estimate is equal to the prior probability: $P(X_t|z_t) = P(X_t|z_{1:t-1})$.

In tracking issues, status includes position, speed, acceleration, bounding boxes that the appearance has detected, and position information provided. This is an example of a multi-camera tracking system based on Bayesian correlation filtering algorithms. Of course, Kalman filtering and particle filtering are also applied. The general idea is similar to Bayesian filtering. The state estimation is performed through observations, and then the model is updated by observation and estimation information.

D. Method based on Cost function

The cost function is another common target tracking method. By designing a reasonable cost function, this kind of method can achieve the tracking trajectory correlation by minimizing the cost function. A lot of work has been done to achieve target tracking by manually extracting features and designing the

cost function carefully. For example, the paper[43;44] depends on the confidence of the detection frame and the spatial timing distance to design this cost function. Zamir et al[45] consider some appearance features, including color histograms, and assigns data associations. In order to solve the problem of tracking sequence association for a longer time, Li et al[46] designed a multi-level association model.

The main approach of multi-camera tracking based on ReID feature and cost function is to use the ReID feature as an appearance model and combine with other state information such as space-time position and moving speed to calculate the relationship matrix between trajectories. Then design a reasonable cost function, use some optimization algorithms to minimize the cost function, and associate the tracking sequences to get the final target's trajectory.

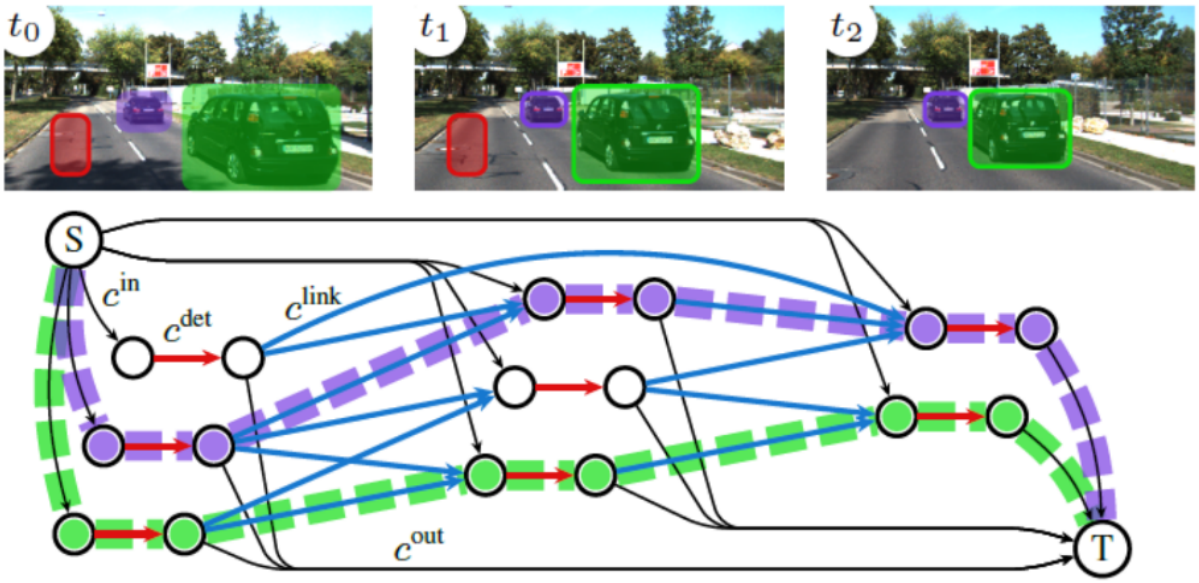

Fig. 22: Example of multi-camera tracking of 3 frames of images using network flow graphs[47]

The work of Schulter et al[47] in 2017 is related to this topic. This paper uses the idea of graph theory to construct a network flow cost function to obtain the tracking trajectory. As shown in 22, the red line in the middle of the two nodes in the figure represents the detection box d_i. This connection evaluates to the stream variable x_i^{det}. For two detection boxes d_i (out node) and d_j (ingress node), satisfy $t(d_i) < t(d_j)$ and $|t(d_i) - t(d_j)| < \tau_t$, The blue arrow in the figure connects the two detected boxes belonging to the same track τ. These lines are assigned to $x_{i,j}^{link}$. The connection does not necessarily need to be a neighboring frame, or it may be a case where occlusion or detection loss is handled across multiple frames. In order to reduce the size of the graph, a limit is imposed on the space of the connection, that is, two detection frames whose spatial positions are far apart are not connected. S and T are the starting and ending points of the track. They are represented by x_i^{in} and x_i^{out} respectively and are connected with other nodes in the figure by using black lines. Each variable in the figure corresponds to a price, and the four variable types correspond to c^{in}, c^{out}, c^{det}, and c^{link}. After that, the problem can be expressed as optimizing the cost function:

$$x^* = \arg\, min c^T x \tag{14}$$

$$s.t. \quad Ax \leq b, Cx = 0 \tag{15}$$

Where $x \in \mathbb{R}^M$ and $c \in \mathbb{R}^M$ are all connection and cost, respectively, and M is the dimension of the problem. In theory, x should be a real integer. To simplify this constraint, they put it to $0 \leq x \leq 1$. The coefficient in the formula (14) is $A = [I, -I]^T \in \mathbb{R}^{2M \times M}$ and $b = [1, 0]^T \in \mathbb{R}^{2M}$. According to the

flow conservation constraint, for $\forall i$, there are $x_i^{in} + \sum_j x_{ji}^{link} = x_i^{det}$ and $x_i^{out} + \sum_j x_{ji}^{link} = X_i^{det}$. And $C \in \mathbb{R}^{2K \times M}$, where K is the number of detected boxes.

We then use the cost function $c(f, \theta)$, where θ is the parameter to be learned and f is the input data. For multi-target tracking problems, The input data includes the coordinates of the detection frame, the confidence level of the detection frame, image characteristics, or some other features. Given some training data that has been marked with a tracking sequence, the final goal is to learn a set of parameters θ to minimize the cost function. So the problem can be turned into an optimization problem:

$$\arg\min_{\theta} L(x^{gt}, x^*) \tag{16}$$

$$S.t. \quad x^* = \arg\min_{x} c(f, \theta)^T x \tag{17}$$

$$Ax \leq b, Cx = 0 \tag{18}$$

The ultimate goal is to minimize the loss function L. By performing some low-order approximations of the objective function, the derivative of the objective function can be obtained to solve the gradient descent method. The specifics can be found in the original paper.

E. Cross-camera matching

The biggest difference between cross-camera tracking and target tracking is that there is one cross-camera matching problem. However, the problem of cross-camera matching is difficult. In addition to using person re-identification to solve the problem, there is little work involved. Here we introduce a work that uses graph theory to solve this problem[48].

Given a set of tracking trajectories T, where T_i^j represents the jth tracking trajectory of the ith camera. We can then construct a graph $G'(V', E', w')$, where each node represents a trace. Suppose we have $\mathcal{I}$ cameras. $A^{i \times j}$ represents the similarity of all tracking trajectories between camera i and camera j. Finally, we can get a similarity matrix:

$$A = \begin{bmatrix} A^{1\times1} & \cdots & A^{1\times j} & \cdots & A^{1\times\mathcal{I}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A^{i\times1} & \cdots & A^{i\times j} & \cdots & A^{i\times\mathcal{I}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A^{\mathcal{I}\times1} & \cdots & A^{\mathcal{I}\times j} & \cdots & A^{\mathcal{I}\times\mathcal{I}} \end{bmatrix}$$

As shown by 23, the color of the node represents the ID of the person, the black line is the trajectory match in the single camera, and the color line is the trajectory match across the camera. We assume that camera 1 contains the trajectory set $\mathcal{Q} = \{T_1^1, T_1^2, T_1^i, T_1^p\}$. $I_{\mathcal{Q}}$ is a diagonal array of $n \times n$, and the diagonal elements are all 1. C_i^j represents the set of tracks generated using the ith tracking trajectory of the jth camera as a constraint set, while C_j refers to the trajectory in the jth camera as the constraint set. The resulting collection, for example, $C_1 = \{C_1^1, C_1^2, C_1^3\}$.

The approximate steps of the algorithm are as follows: $\mathcal{T}$ represents a set of all traced tracks, C represents a set of all track sets, and T_p represents a set of all tracked tracks in the first p cameras. The $\mathcal{F}(\mathcal{Q}, A)$ input constraint set $\mathcal{Q}$ and the relation matrix A output a m local solution $\mathcal{X}^{\backslash \sqcup) \Updownarrow] \int \Updownarrow}$. After that, each set parameter is updated based on this result until all the last traces are clustered. Specific algorithm details can be found in the original literature.

IV. Research Content

The research route is to first study the state-of-art deep neural network based person re-identification and then implement single/multi-camera tracking based on person re-identification technology. This topic

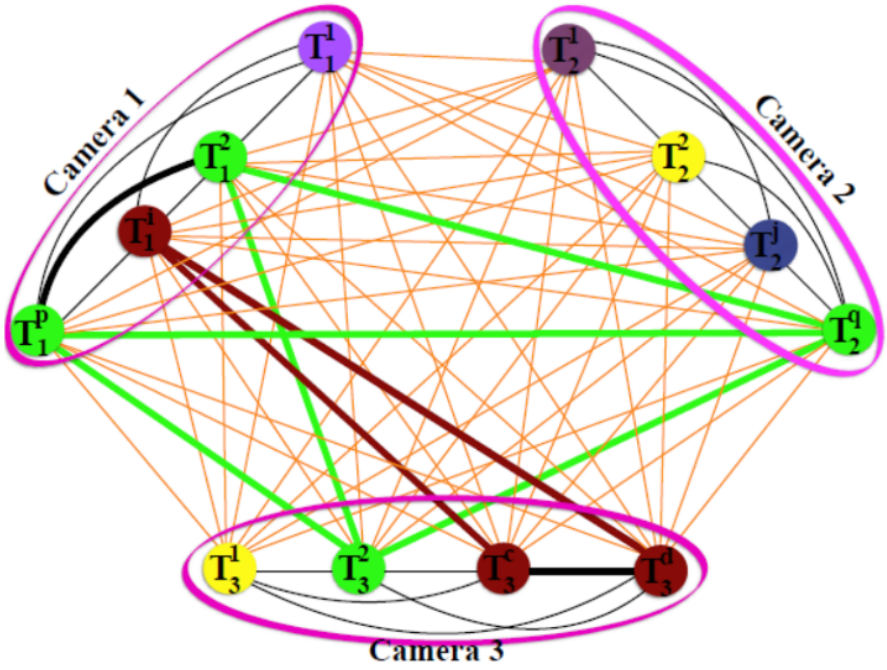

Fig. 23: cross-camera match example[48]

does not include the person detection, we will use the CNN based available open source single shot detectors such as YOLO[49].

Since the nature of the tracking applications often is real-time, the challenge is to develop an algorithm which can perform real-time without much of losing the accuracy. Another part of the research question is to further the algorithm for the cases where there are not enough input training examples for a person.

In order to achieve the research goal, we follow as described below:

1 Implement CNN based re-identification such as YOLO.
2 Use the region proposed by YOLO classifier in order to extract their corresponding features.
3 Extract and examine the CNN features in order to see the amount of information they would provide for re-identification purpose.
4 The later frame of the real-time tracking algorithm is as below, also shows in the figure24 25:

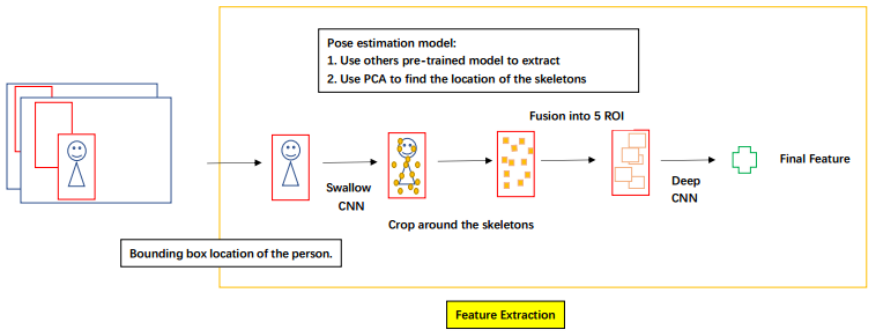

Fig. 24: Framework- Feature extraction

a. Use Triplet Loss[19] as the baseline to calculate the similarity of the person.
b. According to the features extracted by YOLO and the skeleton information we have, use Principle Component Analysis(PCA) as priors combine with a swallow ResNet CNN to find the 14 skeletons

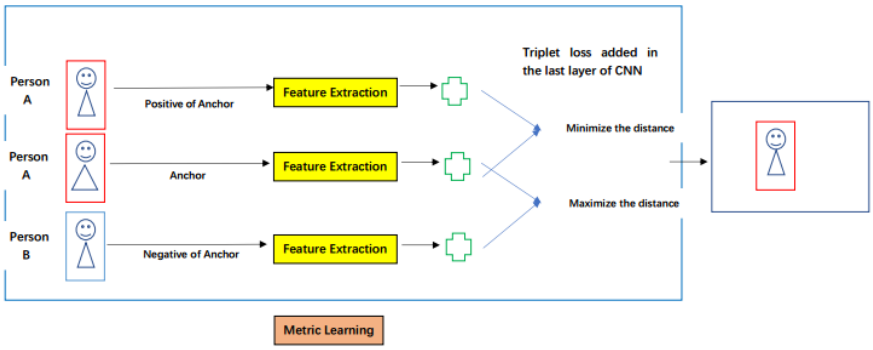

Fig. 25: Framework- Metric learning with Triplet loss

of the person.

c. Crop the input frame images around the skeletons we detected in the last step. And then fusion these 14 skeletons into 5 Regions of Interest and put it in a deep ResNet CNN to extract the feature.

d. According to the Triplet loss, minimize the distance between the anchor and positive atom we choose, maximize the distance between the anchor and the negative atom.

5 After pontificating the person, associated his/her bounding box to achieve the tracking.

REFERENCES

[1] Zheng L, Shen L, Tian L, et al. Scalable person re-identification: A benchmark[C]//Proceedings of the IEEE International Conference on Computer Vision. 2015: 1116-1124.

[2] Zheng L, Bie Z, Sun Y, et al. Mars: A video benchmark for large-scale person re-identification[C]//European Conference on Computer Vision. Springer, Cham, 2016: 868-884.

[3] Li W, Zhao R, Xiao T, et al. Deepreid: Deep filter pairing neural network for person re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 152-159.

[4] Xiao T, Li S, Wang B, et al. End-to-end deep learning for person search[J]. arXiv preprint.

[5] Ristani E, Solera F, Zou R, et al. Performance measures and a data set for multi-target, multi-camera tracking[C]//European Conference on Computer Vision. Springer, Cham, 2016: 17-35.

[6] Gray D, Brennan S, Tao H. Evaluating appearance models for recognition, reacquisition, and tracking[C]//Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS). Citeseer, 2007, 3(5): 1-7.

[7] Hirzer M, Beleznai C, Roth P M, et al. Person re-identification by descriptive and discriminative classification[C]//Scandinavian conference on Image analysis. Springer, Berlin, Heidelberg, 2011: 91-102.

[8] Geng M, Wang Y, Xiang T, et al. Deep transfer learning for person re-identification[J]. arXiv preprint arXiv:1611.05244, 2016.

[9] Lin Y, Zheng L, Zheng Z, et al. Improving person re-identification by attribute and identity learning[J]. arXiv preprint arXiv:1703.07220, 2017.

[10] Zheng L, Yang Y, Hauptmann A G. Person re-identification: Past, present and future[J]. arXiv preprint arXiv:1610.02984, 2016.

[11] Matsukawa T, Suzuki E. Person re-identification using cnn features learned from combination of attributes[C]//Pattern Recognition (ICPR), 2016 23rd International Conference on. IEEE, 2016: 2428-2433.

[12] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[13] Varior R R, Haloi M, Wang G. Gated siamese convolutional neural network architecture for human re-identification[C]//European Conference on Computer Vision. Springer, Cham, 2016: 791-808.
[14] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 815-823.
[15] Liu H, Feng J, Qi M, et al. End-to-end comparative attention networks for person re-identification[J]. arXiv preprint arXiv:1606.04404, 2016.
[16] Cheng D, Gong Y, Zhou S, et al. Person re-identification by multi-channel parts-based cnn with improved triplet loss function[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1335-1344.
[17] Chen W, Chen X, Zhang J, et al. Beyond triplet loss: a deep quadruplet network for person re-identification[C]//Proc. CVPR. 2017, 2.
[18] Liu H, Tian Y, Yang Y, et al. Deep relative distance learning: Tell the difference between similar vehicles[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2167-2175.
[19] Hermans A, Beyer L, Leibe B. In defense of the triplet loss for person re-identification[J]. arXiv preprint arXiv:1703.07737, 2017.
[20] Xiao Q, Cao K, Chen H, et al. Cross domain knowledge transfer for person re-identification[J]. arXiv preprint arXiv:1611.06026, 2016.
[21] Varior R R, Shuai B, Lu J, et al. A siamese long short-term memory architecture for human re-identification[C]//European Conference on Computer Vision. Springer, Cham, 2016: 135-153.
[22] Zheng L, Huang Y, Lu H, et al. Pose invariant embedding for deep person re-identification[J]. arXiv preprint arXiv:1701.07732, 2017.
[23] Zhao H, Tian M, Sun S, et al. Spindle net: Person re-identification with human body region guided feature decomposition and fusion[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 1077-1085.
[24] Wei L, Zhang S, Yao H, et al. Glad: Global-local-alignment descriptor for person re-trieval[C]//Proceedings of the 2017 ACM on Multimedia Conference. ACM, 2017: 420-428.
[25] Zhang X, Luo H, Fan X, et al. Alignedreid: Surpassing human-level performance in person re-identification[J]. arXiv preprint arXiv:1711.08184, 2017.
[26] Wang T, Gong S, Zhu X, et al. Person re-identification by discriminative selection in video ranking[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 38(12): 2501-2514.
[27] Zhang D, Wu W, Cheng H, et al. Image-to-video person re-identification with temporally memorized similarity learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017.
[28] You J, Wu A, Li X, et al. Top-push video-based person re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1345-1353.
[29] Ma X, Zhu X, Gong S, et al. Person re-identification by unsupervised video matching[J]. Pattern Recognition, 2017, 65: 197-210.
[30] McLaughlin N, del Rincon J M, Miller P. Recurrent convolutional network for video-based person re-identification[C]//Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. IEEE, 2016: 1325-1334.
[31] Zhao R, Oyang W, Wang X. Person re-identification by saliency learning[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(2): 356-370.
[32] Liu H, Jie Z, Jayashree K, et al. Video-based person re-identification with accumulative motion context[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017.
[33] Song G, Leng B, Liu Y, et al. Region-based Quality Estimation Network for Large-scale Person Re-identification[J]. arXiv preprint arXiv:1711.08766, 2017.
[34] Zheng Z, Zheng L, Yang Y. Unlabeled samples generated by gan improve the person re-identification baseline in vitro[J]. arXiv preprint arXiv:1701.07717, 2017, 3.
[35] Zhong Z, Zheng L, Zheng Z, et al. Camera Style Adaptation for Person Re-identification[J]. arXiv

preprint arXiv:1711.10295, 2017.
[36] Wei L, Zhang S, Gao W, et al. Person Transfer GAN to Bridge Domain Gap for Person Re-Identification[J]. arXiv preprint arXiv:1711.08565, 2017.
[37] Qian X, Fu Y, Wang W, et al. Pose-Normalized Image Generation for Person Re-identification[J]. arXiv preprint arXiv:1712.02225, 2017.
[38] Ristani E, Solera F, Zou R, et al. Performance measures and a data set for multi-target, multi-camera tracking[C]//European Conference on Computer Vision. Springer, Cham, 2016: 17-35.
[39] Milan A, Leal-TaixÃl' L, Reid I, et al. MOT16: A benchmark for multi-object tracking[J]. arXiv preprint arXiv:1603.00831, 2016.
[40] Kisku D R, Tistarelli M, Sing J K. Computer Vision and Pattern Recognition Workshops[J]. Miami, Florida, USA, 2009: 60.
[41] Beyer L, Breuers S, Kurin V, et al. Towards a Principled Integration of Multi-Camera Re-Identification and Tracking through Optimal Bayes Filters[J]. arXiv preprint arXiv:1705.04608, 2017.
[42] Leal-TaixÃl' L, Canton-Ferrer C, Schindler K. Learning by tracking: Siamese CNN for robust target association[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016: 33-40.
[43] Leal-TaixÃl' L, Pons-Moll G, Rosenhahn B. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker[C]//Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011: 120-127.
[44] Milan A, Roth S, Schindler K. Continuous energy minimization for multitarget tracking[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 36(1): 58-72.
[45] Zamir A R, Dehghan A, Shah M. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs[M]//Computer VisionâĂŞECCV 2012. Springer, Berlin, Heidelberg, 2012: 343-356.
[46] Li Y, Huang C, Nevatia R. Learning to associate: Hybridboosted multi-target tracker for crowded scene[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 2953-2960.
[47] Schulter S, Vernaza P, Choi W, et al. Deep Network Flow for Multi-Object Tracking[J]. arXiv preprint arXiv:1706.08482, 2017.
[48] Tesfaye Y T, Zemene E, Prati A, et al. Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets[J]. arXiv preprint arXiv:1706.06196, 2017.
[49] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.