SMILEtrack: SiMIlarity LEarning for Multiple Object Tracking

[1]Yu-Hsiang Wang ,[1]Jun-Wei Hsieh, [2]Ping-Yang Chen, and [3]Ming-Ching Chang
[1]College of Artificial Intelligence and Green Energy, National Yang Ming Chiao Tung University, Taiwan.
[2]Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan.
[3]College of Artificial Intelligence and Green Energy, University at Albany, SUNY, USA.
[j122333221.ai09,jwhsieh,pingyang.cs08]@nycu.edu.tw, mchang2@albany.edu

Abstract

Multiple Object Tracking (MOT) is widely investigated in computer vision with many applications. Tracking-By-Detection (TBD) is a popular multiple-object tracking paradigm. TBD consists of the first step of object detection and the subsequent of data association, tracklet generation, and update. We propose a Similarity Learning Module (SLM) motivated from the Siamese network to extract important object appearance features and a procedure to combine object motion and appearance features effectively. This design strengthens the modeling of object motion and appearance features for data association. We design a Similarity Matching Cascade (SMC) for the data association of our SMILEtrack tracker. SMILEtrack achieves 81.06 MOTA and 80.5 IDF1 on the MOTChallenge and the MOT17 test set, respectively.

1. Introduction

MOT is a hot topic in computer vision and plays an essential role in video understanding. The goal of MOT is to estimate the trajectories of each target and try to associate them with each frame in video sequences. With the success of MOT, it can be commonly used in society, such as vehicle computing, computer interaction [25] [12], smart video analysis, and autonomous driving. The dominant and efficient MOT strategies [1] [27] [26] are based on the Tracking-By-Detection (TBD) paradigm method in the past few years. It involves tracking according to detection results, which breaks the problem into two steps: detection and association. In the detection step, we need to locate the object of interest in a single video frame, link each object to the existing tracks or create new tracks in the association step. Nevertheless, it still faces challenges due to vague objects, occlusion, and complex scenes.

To accomplish the tracking system, the solution model can be divided into the Separate Detection and Embedding model (SDE) and Joint Detection and Embedding model (JDE). Our method belongs to SDE; the architecture is shown in Figure 1. The SDE requires at least two function components: a detector and a re-identification model. First, the detector locates all the objects in a single frame via bounding boxes. Then the re-identification model will extract the object's features from each bounding box to generate embedding. Finally, associate each bounding box to one of the existing trajectories or create a new track. However, the SDE method cannot achieve real-time inference speed because it requires multiple computations when using two separate models to detect objects and extract embedding. The feature between the detector and re-identification model cannot be shared and the SDE method needs to apply the re-identification model to each bounding box for extract embedding while inference time. Faced with this problem, a feasible solution is to integrate the detector and re-identification models. The JDE category [26] [32] combines the detector and embedding model in a single-shot deep network. It can simultaneously output the detection results and the corresponding appearance embeddings of the detected boxes by only inference the model once.

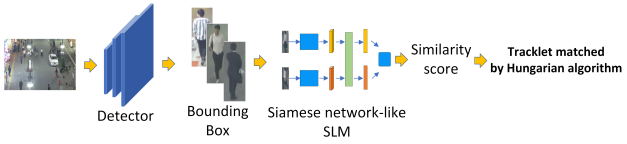

Figure 1. The pipeline of SMILEtrack. SMILEtrack is a Siamese network-like architecture used to learn the object features.

Although the success of the JDE makes the MOT task achieve great accuracy results, we argue that there are still some problems with the JDE. For example, the features conflict between different components. We consider that the features which are needed for object detection tasks and object re-identification tasks are totally different. The features for object detection tasks need high-level features to recognize which classes the object is, but the features for re-identification tasks require more low-level features to dis-

tinguish different instances for the same class. Thus, the shared feature model in JDE could lower the performance of each task. However, as the disadvantage in JDE we mentioned above, the SDE can overcome the shortcomings and still has excellent potential in MOT.

Recently, the Transformer [24] based on the attention mechanism [24] has been introduced into the computer vision field and achieved excellent results. In MOT problems, most of the transformer-based methods use the CNN + transformer framework. It means that the model first extracts the input image feature by a CNN architecture and then shapes those feature maps as input into a transformer. Unlike the tracking-by-detection methods, transformer-based methods achieve the tracking result by joining the detection and data association parts together. It can directly output the track's identity and location by a single model without using any additional tracklet matching skill. Although the transformer-based methods have an outstanding result on feature attention, it still has some limit on the inference speed while inputting the entire image into a transformer architecture.

To generate a high-quality detection and object appearance, we choose the SDE which is a TBD model to solve the features conflict problem in the JDE. However, we argue that most of the feature descriptors cannot distinguish the appearance feature between different objects clearly. To solve this problem, we propose SMILEtrack which combines a detector and a siamese network-like Similarity Learning Module (SLM). Inspired by the vision transformer [6], we create a Image Slicing Attention Block (ISA) that uses the attention mechanism and image slicing mechanism in SLM. Also, we create a Similarity Matching Cascade SMC for matching the object between each frame in the video. The rough process of our tracking system is as follows: First, we predict the target bounding box location by an detector called PRB [4]. After having the object bounding box, we associate the bounding boxes with tracks by the SMC.

The contributions of our work are summarized as follows:

- We introduce a Separate Detection and Embedding model, named SMILEtrack, and the Similarity Learning Module (SLM) which uses a Siamese network-like architecture to learn the similarity between each object.
- For the feature extracting part in SLM, we built an Image Slicing Attention Block (ISA) which uses the image slicing method and the attention mechanism of the transformer to learn the object feature.
- To accomplish the tracklets matching part, we built a Similarity Matching Cascade (SMC) for the step of associating each bounding box in each frame.

2. Related Work

2.1. Tracking-by-Detection

TBD-based algorithms have achieved considerable success in MOT problems, and it has been the most popular way in the MOT framework. The main task of the TBD method is to associate the detection result between each frame in the video to accomplish the MOT system. The whole work can be roughly separated into two parts.

2.1.1 Detection method

Faster R-CNN [18] is a two-stage detector; it uses VGG-16 as the backbone, region proposal network (RPN) for detecting bounding boxes. SSD [11] uses an anchor mechanism to replace RPN; it sets a different size of anchor on each feature map to enhance detection quality. The YOLO series [15] [16] [17] [2] is a one-stage method that uses the feature pyramid network (FPN) to solve the multi-scales problems in object detection, and has an outstanding performance on speed and accuracy. Although the anchor-based detector can achieve an excellent performance, there is still some issue that is caused by anchors. For instance, the anchor-based detector is hard to adjust some hyperparameters for anchor by cases, and it takes a lot of time and memory to calculate the Intersection Over Union (IOU) of the anchor during the training part. In order to overcome these problems, anchor-free detectors are another choice. The CornerNet [9] is an anchor-free method; it utilizes heatmap and corner pooling instead of anchor to predict the top-left and bottom-right corner of the targets, then matches the two points to generate the bounding box for the object. Compared to CornerNet, the CenterNet [34] directly predicts the object's center point by center pooling and cascade corner pooling. YOLOX turns the YOLO series from an anchor-based detector to an anchor-free detector. Also, it uses decoupled heads to improve the accuracy of detection.

2.1.2 Data association method

In the MOT system, many challenges must be conquered, such as object occlusion, crowded scenes, and motion blur. Therefore, the method of data association needs to be treated carefully. SORT [1] first uses the Kalman filter to predict the future location of the object according to the object position at the current frame, then generates the assignment cost matrix via calculating the IOU distance between detection and predicted bounding boxes from the existing targets. Finally, match the assignment cost matrix by the Hungarian algorithm. Although SORT achieves a high-speed inference time, it cannot handle the long-term occlusion problem or a fast motion object because it doesn't concern the object appearance information.

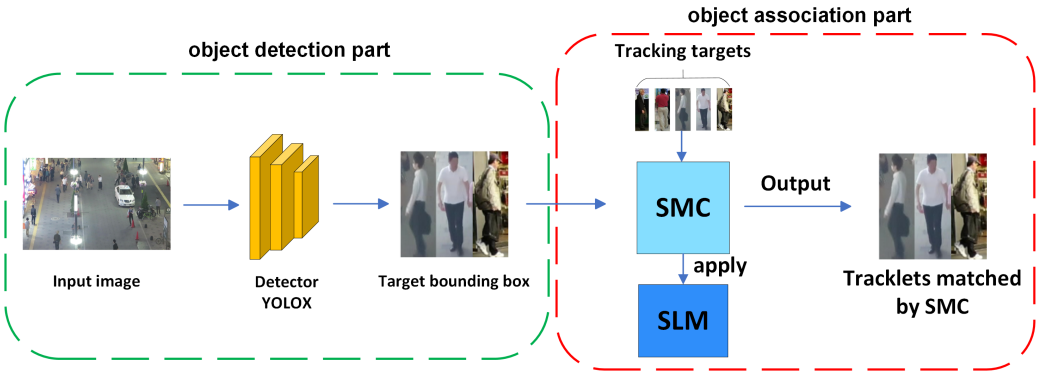

Figure 2. The SMILEtrack architecture consists of two parts: (1) object detection and (2) object association.

To solve the occlusion problem, Deep SORT [27] applies a pre-trained CNN model to extract the bounding box appearance feature, then uses the appearance feature to compute the similarity between tracklets and detections. Finally uses the Hungarian algorithm to accomplish the assignment. This way can efficiently reduce the number of ID switches, but the detection model and the feature extract model are separate in Deep SORT, which causes the inference speed to be far from real-time. Face this problem, JDE [26] combines the Detector and Embedding model in a one-shot network, it can run in real-time and is comparably accurate to the two-stage method. FairMOT [32] demonstrates the unfairness caused by anchors, it applies an anchor-free method that is built on top of CenterNet, and it improves the performance by a large margin in several datasets, such as MOT17 [14]. However, we claim that there are some problems in the JDE models, such as the features conflict between different components.

Meanwhile, several MOT tracking methods [21] [22] have discarded the object appearance feature and accomplished the tracking system only by applying high-performance detectors and motion information. Even though these methods could reach state-of-the-art performance and a high inference speed in MOTChallenge benchmarks, we dispute that it's partly due to the simplicity of motion patterns in the MOTChallenge benchmark dataset. Furthermore, not referring the object appearance feature could lead the object tracking accuracy to poor robustness in more crowded scenes.

2.2. Tracking-by-Attention

With success in object detection by using transformers, Trackformer [13] casts the MOT as a set prediction problem, which is based on DETR and adds the object query and autoregressive track queries for object tracking. TransTrack [23] built on Deformable DETR and has two decoders, one for the current frame detection, and another for previous frame detection. It accomplishes the tracking problem by matching the detection box between the two decoders. TransCenter [29] is a point-based tracking that proposes a dense query feature map with a multi-scale of the input image for MOT leveraging transformers.

3. Methodology

In this section we present the details of the SMILEtrack model, including the Similarity Learning Module (SLM) and the Similarity Matching Cascade (SMC) for box association in each frame.

3.1. Architecture Overview

The overall architecture of SMILEtrack is described in Figure 2. Our framework can be divided into the following steps. (1) Detecting object location: To locate the target object position, we apply PRB as the detector. (2) Data association: The MOT problem is achieved by associating each object from adjacent frames. After having the detection result generated by PRB [4], we compute motion affinity matrix and appearance affinity matrix between each frame, then solve the linear assignment problem via Hungarian algorithm with the cost matrix which is combined with these two matrices.

3.2. Similarity Learning Module (SLM) for Re-ID

To achieve a robustly tracking quality, the object appearance information is indispensable. Several tracking methods have taken the object appearance information into account. For example, DeepSORT applies a deep appearance descriptor constructed with a simple CNN to extract the target appearance feature. Although the appearance descriptor could extract a useful appearance feature, we complain that the appearance descriptor cannot distinguish the appearance feature between different objects clearly. To extract more discriminative appearance features, we propose a similarity learning module SLM similar to siamese network architecture. The detail of SLM is shown in Figure 3.

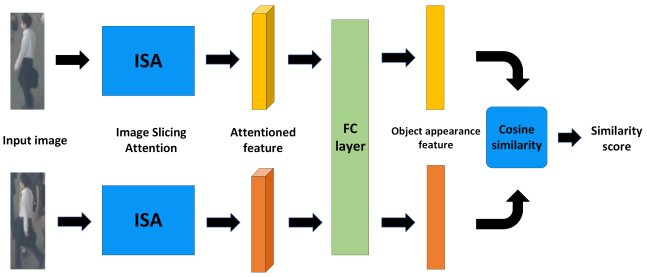

Figure 3. The full architecture of the similarity learning module SLM.

For the input of SLM, we put two different images into the SLM simultaneously. Both of them will pass through the ISA feature extractor in § 3.2.1, which shares parameters between two images. The architecture of ISA will introduce in more detail later. After extracting the feature of the input image, we use a fully connected layer to integrate the feature. For learning a robust appearance feature that can distinguish various objects, we apply the cosine similarity distance to compute the similarity between the two images. The similarity score between the same objects should be as high as possible; otherwise, the similarity score between the different objects should be close to zero.

3.2.1 The Image Slicing Attention (ISA) Block

To produce a reliable appearance feature, a superior feature extractor is essential. Although the transformer has an outstanding performance on feature enhancement, we consider that adding the full encoder-decoder architecture into the tracking system is too heavy for the model computation and the parameter size. Inspired by VIT, we construct the ISA that applies an image slicing technique and the attention mechanism for feature extraction. The detailed architecture of ISA is shown in Figure 4.

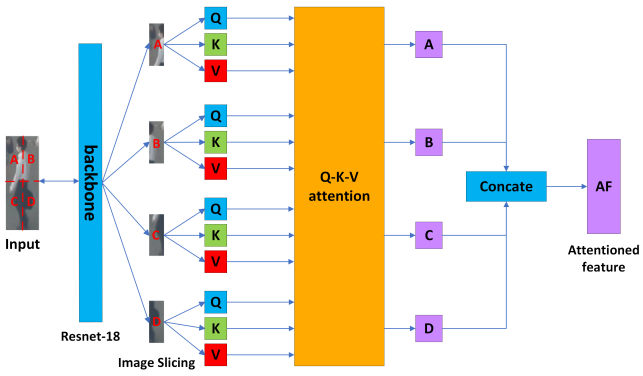

Figure 4. The full architecture of Image Slicing Attention Block – ISA. Applying image slicing for the input image, we divide it to the top-left part, top-right part, bottom-left, bottom-right part.

3.2.2 Image Slicing

The basic transformer receives a 1D-vector as the input of the encoder. For a 2D-image, it will increase the computational complexity by setting the entire image as the input directly. A practical method is dividing the image into slices. For preparing the input of the Q-K-V attention block [24], we generate the bounding box of the object via the detector first. Since each of the bounding boxes has different scales, we resize them to a fixed size $B \in R^{w \times h}$, where (w, h) are the width and height of the region proposal. Notice that we set $w = 80$, $h = 224$ while training the MOT dataset. To generate the feature map of the resized bounding box, we apply the backbone Resnet-18 [8] to extract the feature. After having the feature map of the bounding box, we divide it into slices $S_i \in R^{n \times s \times t}$ of size $s \times t$, where $n = 4$ is the number of slices. Furthermore, we add a $1D$ position embedding E_P to each slice. Each slice can be represented as the following equation:

$$S_i = S_i + E_P, \quad i = A, B, C, D, \quad E_P = 1, 2, 3, 4 \quad (1)$$

Eventually, We apply the feature slices sequence $S = \{S_A \sim S_D\}$ as the input of the attention block.

3.2.3 The Q-K-V attention block

The standard transformer is adept at handling long-term complex dependencies between sequences, such as natural language processing. The most significant part is the attention block in the transformer. The transformer computes the attention function by packing queries into a matrix Q, also the keys and values are packed into matrices K and V. The calculation of the attention block is expressed as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (2)$$

where d_k is the dimension of the key vector. To generate the queries, keys, and values for the attention block, we apply a one-by-one fully connection layer for each slice produced by Image Slicing. Each slice has an output S_i after passing the Q-K-V attention block. We denote the output of each slice $S = \{S_A \sim S_D\}$ through the Q-K-V attention block as the following equation:

$$\begin{aligned} S_A &= SA(Q_{S1}, K_{S1}, V_{S1}) + CA(Q_{S1}, K_{S2}, V_{S2}) \\ &\quad + CA(Q_{S1}, K_{S3}, V_{S3}) + CA(Q_{S1}, K_{S4}, V_{S4}) \\ S_B &= SA(Q_{S2}, K_{S2}, V_{S2}) + CA(Q_{S2}, K_{S1}, V_{S1}) \\ &\quad + CA(Q_{S2}, K_{S3}, V_{S3}) + CA(Q_{S2}, K_{S4}, V_{S4}) \\ S_C &= SA(Q_{S3}, K_{S3}, V_{S3}) + CA(Q_{S3}, K_{S1}, V_{S1}) \\ &\quad + CA(Q_{S3}, K_{S2}, V_{S2}) + CA(Q_{S3}, K_{S4}, V_{S4}) \\ S_D &= SA(Q_{S4}, K_{S4}, V_{S4}) + CA(Q_{S4}, K_{S1}, V_{S1}) \\ &\quad + CA(Q_{S4}, K_{S2}, V_{S2}) + CA(Q_{S4}, K_{S3}, V_{S3}) \end{aligned} \quad (3)$$

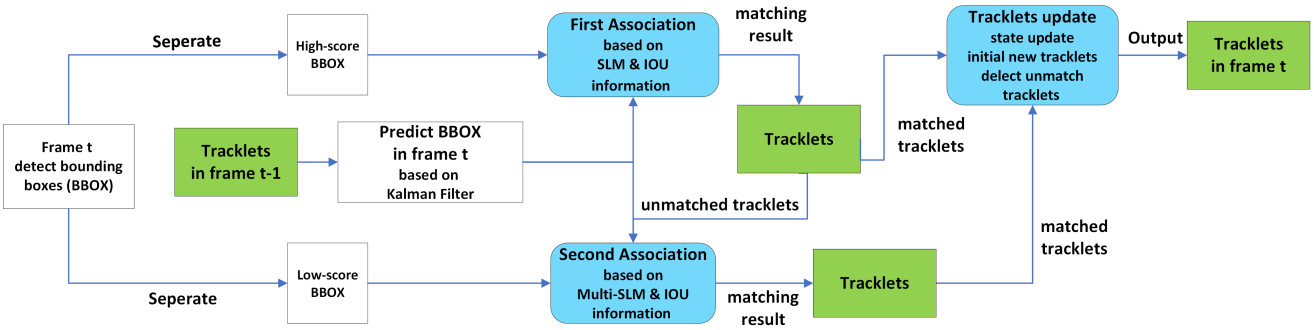

Figure 5. The pipeline of Similarity Matching Cascade (SMC).

where Q_{Si} is the Query matrix which is obtained by S_i, K_{Si} is the Key matrix which is obtained by S_i and V_{Si} is the Value matrix which is obtained by S_i. The SA is the Self-Attention. The CA is the Cross-Attention. Both of the Self-Attention and Cross-Attention are calculated as equation 2. After having the feature $S = \{S_1 \sim S_4\}$, we use the concatenate mechanism to fuse them so as to retain the features of the input image.

3.3. Similarity Matching Cascade (SMC) for Target Tracking

The object association part is crucial to the tracking-by-detection paradigm method. Choosing different strategies for the matching part will lead to entirely different results. ByteTrack is a simple, effective association method. It keeps each detection box and divides them into high and low confidence score ones, then associates them with IOU distance. Although ByteTrack reaches the state-of-art performance in MOT, we dispute that it's partly in view of the simplicity of motion patterns in the MOTChallenge benchmark dataset. It still has some issues if only using the IOU distance information in the association part, such as the id-switch problem will occur when the targets are getting closer. For solving the issue, we design a variant association method by integrating the advantage of ByteTrack and our SLM. The matching pipeline of our method is shown in Figure 5 and the pseudo-code of the association method is shown in the supplementary.

First, we confirm all the detection boxes det_i in the current frame, and divide them into D_{high} set and D_{low} set by thresholds $thres$. For the setting of the $thres$ value, we rearrange the detection d in det_i according to their score from low to high, then compute the mean score of the first half d in $deti$, and set the mean score to $thres$. After we have the thresholds $thres$, we put the detection box whose score is higher than $thres$ into D_{high}, and put the detection box whose score is between $thres$ and 0.1 into D_{low}. We regarded the detection box whose score is lower than 0.1 as background or noise. After separating the detection boxes, we fused the lost object list LL to the tracking list TL, and used Kalman filter to predict each object position at the current frame in TL. The association part is mainly divided into two stages.

(Stage I) In the first association stage, we focus on the D_{high} set first. We calculate the motion matrix M_m and appearance similarity matrix M_a of D_{high} and TL. For the motion matrix M_m, we compute the IOU distance between TL and D_{high}. For the appearance similarity matrix M_a, it is computed by the SLM. Then we fuse the matrix M_m and M_a as cost matrix C_{high} by the $Gate$ function that we purpose:

$$C_{high} = M_m(i,j) - (1 - M_a(i,j)) \tag{4}$$

where $M_m(i,j)$ is the IOU distance between the i-th tracklet and the j-th detection, and the $M_a(i,j)$ is the feature similarity between the i-th tracklet and the j-th detection that is generated by SLM. Finally, complete the linear assignment by Hungarian algorithm with cost matrix C_{high} in the first stage matching. The unmatched detection of D_{high} and the unmatched tracks of TL are put in D_{Remain} and TL_{Remain}.

(Stage II) In the second matching stage, we match the D_{low} and TL_{Remain}. The motion matrix M_m of D_{low} and TL_{Remain} is calculated the same as the first matching stage. For the appearance similarity matrix M_a, we build a multi-template-SLM for learning the similarity between the low score detection and tracks. While handling the low score detection, using the feature of tracks in the last frame directly to compute the similarity may obtain an unreliable score because the low score detection object feature is different from the tracks that are caused by some occlusion. To fight this issue, we apply a feature bank mechanism for saving the track's various features in different frames. The similarity score between the feature bank F_i of the i-th track and the low score j-th detection is computed as:

$$M_a(i,j) = max\left\{SLM(fi,dj) \mid \text{for all } f_i \in F_i\right\} \tag{5}$$

After having the matrix M_m and M_a, we generate the cost matrix C_{low} by fusing the matrix M_m and M_a as the same

as the first matching stage and complete the linear assignment by Hungarian algorithm with cost matrix C_{low}. The unmatched detection of D_{low} and the unmatched tracks of TL_{Remain} are put in $D_{RRemain}$ and $TL_{RRemain}$.

Afterward finishing the object association stage, we set a threshold H for initializing new tracks. The unmatched detection in D_{Remain} whose score is higher than H can initialize a new track, and move the unmatched tracks in $TL_{RRemain}$ to the lost object list LL. We regard the unmatched detection $D_{RRemain}$ as background. Notice that we delete the tracks in LL only when the tracks exist more than 30 frames in LL.

4. Experimental Results

4.1. Dataset

We conduct experiments on the MOTChallenge [14] benchmark. Specifically, evaluations are performed on the MOT17 [14] test set following the "private detection" protocol. MOT17 is the most popular dataset in MOTChallenge. It contains 14 video sequences (7 sequences for training and the other 7 for testing) with both moving and static cameras.

Other common MOT datasets include ETH [19], CalTech [5], MOT16 [14], CityPerson [31], CrowdHuman [20], ETHZ [7], CUHK-SYSU [28] and PRW [33]. The ETH, MOT16, and CityPerson datasets only provide bounding box annotations for training detection models; thus additional datasets are required for the case of training re-ID and MOT model. The CalTech, PRW, and CUHK-SYSU datasets provide both the bounding box locations and identity annotations for training re-ID models.

We train SMILEtrack on the combination of the MOT17 training set, CrowdHuman, ETHZ, and Cityperson. For ablation studies, we use the first half of the training set for model training, use the other half for validation. The pedestrian images from the MOT17 video sequences are cropped for training our SLM re-ID model.

4.2. MOT Evaluation Metrics

Standard MOT evaluation metrics include the Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Identity F1 Score (IDF1), Mostly Tracked (MT), Mostly Lost (ML), False Positive Rate (FP), False Negative Rate (FN), ID Precise (IDP), and ID switches (IDs). Out of these, the MOTA and IDF1 are two most commonly used metrics. The formula of MOTA and IDF1 is shown in equation 6 and 7.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \quad (6)$$

$$IDF_1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \quad (7)$$

MOTA is a combination of FP, FN, and IDs to reflect the detection performance. In contrast, IDF1 focuses more on identity matching ability and data association performance.

4.3. Implementation Details

We train the PRB [4] detector on the COCO [10] dataset for weight initiation; the model is fine-tuned on both MOT16 and MOT17 datasets to improve the person detection performance. We apply several data augmentation methods including Mosaic [2] and Mixup [30] during training.

For the evaluation on MOT17, we train PRB for 100 epochs on the combination of MOT17 training set, CrowdHuman, ETHZ, and Cityperson. The size of the input image is 1440 $\times$ 800. We choose the SGD optimizer and set the initial learning rate as 10^{-3} with cosine annealing schedule.

For training the SLM, we train on our own dataset that crops from the MOT17 training set. Since each of the pedestrians that crop from MOT17 has a different size, we resize the pedestrian to a fixed size 224 $\times$ 80. We choose the optimizer SGD and the learning rate is initial to 6.5×10^{-3} with cosine annealing schedule. We train for 150 epochs with MSE [3] loss.

For the Gate function we propose in SMC, we set the threshold ε to 0.7 for filtering the cost matrix. At the linear assignment stage, we reject the matching whose cost matrix between the detection and the tracks is higher than 0.2. For initial new objects, the threshold H is set to 0.7 for filtering the detection. For the feature bank which is used in the multi-template-SLM for each object, we set the feature bank size to be able to store 50 frame appearance. Furthermore, we divide the feature bank into two categories, high-score template and low-score template. For those detections which has a high confidence score, we store the detection appearance feature into the high-score template; otherwise we put the appearance feature of low confidence score detection into a low-score template.

4.4. Evaluation Results

Table 1 shows the evaluation result of SMILEtrack against the state-of-the-art trackers on the MOT17 test set following the "private detector" protocol of the MOTChallenge. All results are generated using the official MOTChallenge evaluation website. SMILEtrack achieves an outstanding result of 80.3 MOTA and 77.3 IDF1. Specifically, we use the best results from the ablation study as the model setting for MOT17 evaluation. We set the SLM similarity feature dimension to 256 for a tracked target. The IOU and appearance information is used calculate the similarity matrix for the two SMC matching stages. Gate function is applied to fuse the IOU and appearance information, and multi-template-SLM is used in addressing the issue of low detection scores.

Table 1. Comparison against the state-of-the-art methods under the "private detector" protocol on the MOT17 [14] test set.

Method	MOTA ↑	IDF1 ↑	FN ↓	FP ↓	IDs ↓	MT ↑	ML ↓
DAN	52.4	49.5	234592	25423	8431	21.4%	30.7%
TubeTK	63.0	58.6	177483	27060	4137	31.2%	19.9%
CenterTrack	67.8	64.7	160332	**18498**	3039	34.6%	24.6%
MOTR	65.1	66.4	149307	45486	2049	33.0%	25.2%
QuasiDense	68.7	66.3	146643	26589	3378	40.6%	29.1%
MAT	69.5	63.1	138741	30660	2844	43.8%	18.9%
SOTMOT	71.0	71.9	118983	39537	5184	42.7%	15.3%
FairMOT	73.7	72.3	117477	27507	3303	43.2%	17.3%
CSTrack	74.9	72.6	114303	23847	3567	41.5%	17.5%
TransTrack	75.2	63.5	86442	50157	3603	55.3%	**10.2**%
CorrTracker	76.5	73.6	99510	29808	3369	47.6%	12.7%
BQTQ	77.7	74.5	100908	22401	2631	42.4%	15.4%
CountingSORT	78.0	74.8	92247	28233	3453	49.8%	15.4%
StrongSORT	79.6	79.5	86205	27876	**1194**	53.6%	13.9%
ByteTrack	80.3	77.3	83721	25491	2196	53.2%	14.5%
BoT-SORT	80.6	79.5	85398	22524	1257	-	-
SMILEtrack(Ours)	**81.06**	**80.5**	**82682**	22963	1246	**53.6**%	14.7%

Table 2. Performance regarding feature dimensions.

Feature dim	MOTA ↑	IDF1 ↑	IDS ↓
64	**76.5**	78.1	**621**
128	76.4	**78.4**	633
256	76.3	78.1	645

4.5. Ablation study

We perform ablation study using the same weights from the training of the combination of the MOT17 validation set and the CrowdHuman dataset. This ensures fairness and prevents the impact of detector variances.

4.5.1 Similarity feature dimension

The selection of feature dimension in representing the pedestrian ($i.e.$ the object appearance size in SLM) can greatly affect the MOT accuracy, and the settings for detection and re-ID features are not be the same. For detection, the feature dimension is usually preferred to be larger, since target detection requires abundant high-level features. In comparison, the re-ID features require more low-level appearance features to discriminate among candidates.

We test different object appearance sizes in SLM and the result is shown in Table 2. Observe that the feature dimension of 64 leads to the best MOTA and the IDs, while the dimension of 128 leads to the best IDF1. We found that the performance of MOTA and IDs improve as the feature dimension decreases. We choose dimension 128 for the object appearance size in SLM, which maximizes the overall MOTChallenge performance.

4.5.2 Similarity matrix

For data association, the similarity matrix between the detection and tracks is the key factor for matching objects. Most methods select IOU information or appearance information for the similarity matrix. In our SMC, the main association consists of two stages. We evaluate the combination of the IOU information or the appearance information for the similarity matrix for stage I and stage II. The result is shown in Table 4. Notice that the SLM means the appearance information of the object. The combination of the IOU information and appearance information follows the equation:

$$\text{Similarity matrix} = \alpha \cdot IOU + (1 - \alpha) \cdot SLM, \quad (8)$$

where the weighting parameter α is set to 0.5.

Compared with row 1 and row 2 in Table 4, the similarity matrix in stage 1 with the combination of IOU information and appearance information achieves a higher MOTA and IDF1 that uses IOU information only. Observe that in row 1 and row 3, the use of appearance information on the low score detection boxes results in a lower MOTA and IDF1. The reason is that the low score detection boxes usually include some occlusion or motion blur that makes the appearance information unreliable. In Table 4, the best result is obtained using both IOU and appearance for stage 1 and only using IOU for stage 2.

4.5.3 Appearance matching using gate function and Multi-template-SLM

We perform ablation study on the gate function and Multi-template-SLM. Table 3 shows the results. The IOU and ap-

Table 3. Ablation study on the MOT17 validation set for different strategies.

Similarity matrix for stage1	Similarity matrix for stage 2	Gate function	Multi-template-SLM	MOTA	IDF1	IDS
SLM w/ IOU	SLM w/ IOU			76.4	77.9	663
SLM w/ IOU	SLM w/ IOU	√		76.5	78.3	621
SLM w/ IOU	SLM w/ IOU		√	76.5	78.5	615
SLM w/ IOU	SLM w/ IOU	√	√	**76.6**	**79.2**	**545**

Table 4. Comparison of different strategies in Stages 1 & 2 on the MOT17 validation set.

Similarity matrix for stage 1	Similarity matrix for stage 2	MOTA ↑	IDF1 ↑	IDs ↓
IOU	IOU	76.2	74.0	731
SLM w/ IOU	IOU	**76.5**	**78.8**	**585**
IOU	SLM w/ IOU	76.1	73.7	740
SLM w/ IOU	SLM w/ IOU	76.4	77.9	663

(a) Matching without gate function.

(b) Matching with gate function.

Figure 6. The comparison between the different strategies for fusing IOU and appearance information. The top-left number of each bounding box represent the target ID. When the two targets are getting closer and the IOU score is higher than the appearance score, using the common weighted sum of the IOU and appearance information may cause an ID-switch problem. Our proposed Gate function prevents the ID-switch problem from occurring.

pearance is effective information for matching. Most of the methods combine the IOU and appearance information by Eq. (8). This way may cause problems when the IOU score is much higher than the appearance similarity score between two different pedestrians. To solve this problem, we propose gate function to reject target matching whose appearance similarity score is lower than $\epsilon = 0.7$ even if they have a high IOU score. Faced with the unreliable feature problem in low score detection boxes, we apply the Multi-template-SLM mechanism which uses the feature bank to store the different appearances of the object. We apply the Gate function and Multi-template-SLM in the matching part. For the similarity matrix, we use IOU information and the appearance information for both stage 1 and stage 2. The best performance is applying the Gate function to stage I and II and Multi-template-SLM to stage 2. We show some visualization results of the video MOT17-04-FRCNN in Figure 6.

We find that using the common weighted sum of the IOU and appearance information will cause problems in the case we mention in Figure 6. However, applying the Gate function to fuse the IOU and appearance information can overcome this problem.

5. Conclusion

In this paper, we present SMILEtrack, a Siamese network-like architecture that can effectively learn object appearance for single-camera multiple object tracking. We develope a Similarity Matching Cascade (SMC) for bounding box association in each frame. Experiments show that our SMILEtrack achieves high MOTA, IDF1, and IDs performance scores on MOT17.

Future work. Since the SMILEtrack is a Separate Detection and Embedding (SDE) method, it runs slower than the Joint Detection and Embedding (JDE) methods. In the future, we will investigate approaches that can improve the MOT time *vs.* accuracy trade-off.

References

[1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016. 1, 2

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-yuan Liao. Yolov4: Optimal speed and accuracy of object detection. In *arXiv:2004.10934*, 04 2020. 2, 6

[3] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014. 6

[4] Ping-Yang Chen, Ming-Ching Chang, Jun-Wei Hsieh, and Yong-Sheng Chen. Parallel residual bi-fusion feature pyramid network for accurate single-shot object detection. *IEEE Transactions on Image Processing*, 30:9099–9111, 2021. 2, 3, 6

[5] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009. 6

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 2

[7] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person

tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 4

[9] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 6

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016. 2

[12] Chenxu Luo, Chang Ma, Chunyu Wang, and Yizhou Wang. Learning discriminative activated simplices for action recognition. In *Conference on Artificial Intelligence (AAAI)*, January 2017. 1

[13] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. 2021. 3

[14] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking, 2016. 3, 6, 7

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. 2

[16] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017. 2

[17] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018. cite arxiv:1804.02767Comment: Tech Report. 2

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 2

[19] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[20] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd, 2018. 6

[21] Daniel Stadler and Jurgen Beyerer. On the performance of crowd-specific detectors in multi-pedestrian tracking. In *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, page 1–12. Institute of Electrical and Electronics Engineers (IEEE), 2021. 3

[22] Daniel Stadler and Jurgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 133–142, 2022. 3

[23] Peize Sun, Yi Jiang, Zhang Rufeng, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer, 12 2020. 3

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2, 4

[25] Chunyu Wang, Yizhou Wang, and Alan L. Yuille. An approach to pose-based action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2013. 1

[26] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *The European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[27] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. 1, 3

[28] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search, 2017. 6

[29] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking, 03 2021. 3

[30] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017. 6

[31] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. pages 4457–4465, 07 2017. 6

[32] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. 1, 3

[33] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild, 2017. 6

[34] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 2