Joint Learning of Convolutional Neural Networks and Temporally Constrained Metrics for Tracklet Association

Bing Wang, Li Wang, Bing Shuai, Zhen Zuo, Ting Liu, Kap Luk Chan, Gang Wang
School of Electrical and Electronic Engineering, Nanyang Technological University
50 Nanyang Avenue, 639798, Singapore
{wang0775,wa0002li,bshuai001,zzuo1,liut0016,eklchan,wanggang}@ntu.edu.sg

Abstract

In this paper, we study the challenging problem of multi-object tracking in a complex scene captured by a single camera. Different from the existing tracklet association-based tracking methods, we propose a novel and efficient way to obtain discriminative appearance-based tracklet affinity models. Our proposed method jointly learns the convolutional neural networks (CNNs) and temporally constrained metrics. In our method, a siamese convolutional neural network (CNN) is first pre-trained on the auxiliary data. Then the siamese CNN and temporally constrained metrics are jointly learned online to construct the appearance-based tracklet affinity models. The proposed method can jointly learn the hierarchical deep features and temporally constrained segment-wise metrics under a unified framework. For reliable association between tracklets, a novel loss function incorporating temporally constrained multi-task learning mechanism is proposed. By employing the proposed method, tracklet association can be accomplished even in challenging situations. Moreover, a large-scale dataset with 40 fully annotated sequences is created to facilitate the tracking evaluation. Experimental results on five public datasets and the new large-scale dataset show that our method outperforms several state-of-the-art approaches in multi-object tracking.

1. Introduction

Multi-object tracking in real scenes is an important topic in computer vision, due to its demands in many essential applications such as surveillance, robotics, traffic safety and entertainments. As the seminal achievements were obtained in object detection [10, 40, 14], tracklet association-based tracking methods [21, 44, 11, 41, 37] have become popular recently. These methods usually include two key components: 1) A tracklet affinity model that estimates the linking probability between tracklets (track fragments), which is usually based on the combination of multiple cues (motion and appearance cues); 2) A global optimization framework for tracklet association, which is usually formulated as a maximum a posterior problem (MAP).

Even though some state-of-the-art methods [21, 11, 4] have achieved much progress in constructing more discriminative appearance and motion based tracklet affinity models, problems such as track fragmentation and identity switch still cannot be well handled, especially under difficult situations where the appearance or motion of an object changes abruptly and significantly. Most of state-of-the-art tracklet association-based multi-object tracking methods make use of image representations which are often not well-suited for constructing robust appearance-based tracklet affinity models. Current methods usually utilize pre-selected features, such as HOG features [10], local binary patterns [40], or color histograms, which are not "tailor-made" for the tracked objects in question. Recently, deep convolutional neural network architectures have been successfully applied to many challenging tasks, such as image classification [20] and object detection [16], and reported highly promising results. The core to its success is to take advantage of deep architectures to learn richer hierarchical features through multiple nonlinear transformations. Hence, we adopt the deep convolutional neural network for multi-object tracking in this work.

Traditional deep neural networks are designed for the classification task. Here, we aim to associate tracklets by joint learning of the convolutional neural networks and the appearance-based tracklet affinity models. This joint optimization will maximize their capacity for solving tracklet association problems. Hence, we propose to jointly learn the siamese convolutional neural network, which consists of two sub-networks (see Figure 1), and appearance-based tracklet affinity models, so that the appearance-based affinity models and the "tailor-made" hierarchical features for tracked targets are learned simultaneously and coherently. Furthermore, based on the analysis of the characteristics of the sequential data stream, a novel temporally constrained

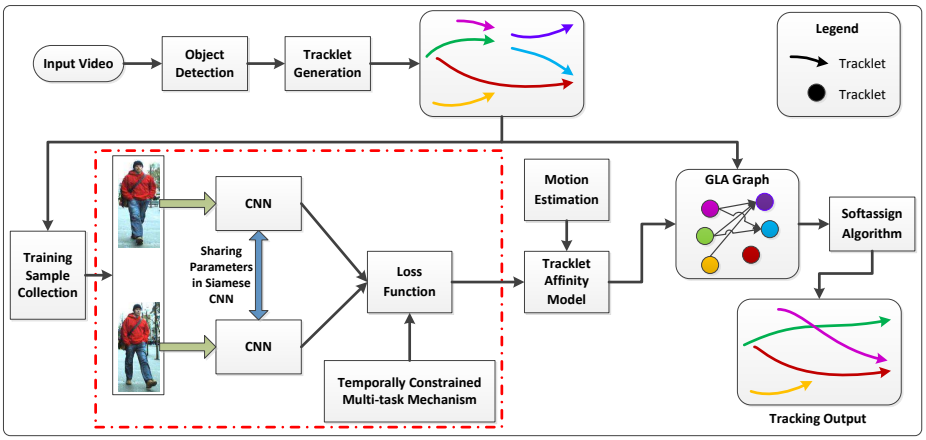

Figure 1. Tracking framework of our method. In the Generalized Linear Assignment (GLA) graph, each node denotes a reliable tracklet; each edge denotes a possible link between two tracklets. We jointly learn the siamese CNN and the temporally constrained metrics for tracklet affinity model, as shown in the red-dashed box, which estimates the linking probability between two tracklets in the GLA graph. The tracking results are obtained by combinatorial optimization using the softassign algorithm.

multi-task learning mechanism is proposed to be added to the objective function. This makes the deep architectures more effective in tackling the tracklet association problem. Although deep architectures have been employed in single object tracking [39, 26, 38, 45, 18], we explore deep architectures for multi-object tracking in this work.

The proposed framework in this paper is shown in Figure 1. Given a video input, we first detect objects in each frame by a pre-trained object detector, such as the popular DPM detector [14]. Then a dual-threshold strategy [19] is employed to generate reliable tracklets. The siamese CNN is first pre-trained on the auxiliary data offline. Subsequently, the siamese CNN and temporally constrained metrics are jointly learned for tracklet affinity models by using the online collected training samples among the reliable tracklets. Finally, the tracklet association problem is formulated as a Generalized Linear Assignment (GLA) problem, which is solved by the softassign algorithm [17]. The final trajectories of multiple objects are obtained after a trajectory recovery process.

The contributions of this paper can be summarized as: (1) We propose a unified deep model for jointly learning "tailor-made" hierarchical features for currently tracked objects and temporally constrained segment-wise metrics for tracklet affinity models. With this deep model, the feature learning and the discriminative tracklet affinity model learning can efficiently interact with each other, maximizing their performance co-operatively. (2) A novel temporally constrained multi-task learning mechanism is proposed to be embedded into the last layer of the unified deep neural network, which makes it more effective to learn appearance-based affinity model for tracklet association. (3) A new dataset with 40 diverse fully annotated sequences is built to facilitate performance evaluation. This new dataset includes 24,882 frames and 246,330 annotated bounding boxes.

2. The Unified Deep Model

In this section, we explain how the unified deep model is designed for jointly learning hierarchical features and temporally constrained metrics for tracklet association.

2.1. The Architecture

A deep neural network usually works in a standalone mode for most of computer vision tasks, such as image classification, object recognition and detection. The input and output of the deep neural network in this mode are a sample and a predicted label respectively. However, for the tracklet association problem, the objective is to estimate the tracklet affinities between two tracklets to decide whether they belong to the same object. Hence, the "sample $\rightarrow$ label" mode deep neural network is not applicable to the tracklet association problem. To deal with this problem, we propose to create a siamese deep neural network, which consists of two sub-networks working in a "sample pair $\rightarrow$ similarity" mode.

The structure of the siamese convolution neural network (CNN) is shown in Figure 1 (red-dashed box). Given two target images, they are first warped to a fixed 96 $\times$ 96 patch and presented to the siamese CNN. The siamese CNN is composed of two sub convolutional neural networks (CNNs), as shown in Figure 1 (red-dashed box). A novel metric learning based loss function is proposed for learning this siamese CNN. Moreover, the siamese CNN has their two sub-CNNs sharing the same parameters, *i.e.*, weights and biases.

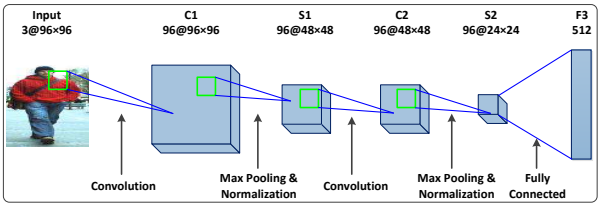

Figure 2. The structure of 5-layer sub-CNN used in the unified deep model.

The sub-CNN in the unified deep model consists of 2 convolutional layers (C1 and C2), 2 max pooling layers (S1 and S2) and a fully connected layer (F3), as shown in Figure 2. The number of channels of convolutional and max pooling layers are both 96. The output of the sub-CNN is a feature vector of 512 dimensions. A cross-channel normalization unit is included in each pooling layer. The convolutional layer output has the same size as the input by zero padding of the input data. The filter sizes of C1 and C2 layers are 7×7 and 5×5 respectively. The activation function for each layer in the CNN is ReLU neuron [20].

2.2. Loss Function and Temporally Constrained Metric Learning

As we can see in Figure 1 (red-dashed box), the siamese CNN consists of two basic components: two sub-CNNs and a loss function. The loss function converts the difference between the input sample pair into a margin-based loss.

The relative distance between an input sample pair used in the loss function, parameterized as a Mahalanobis distance, is defined as:

$$\|x_i - x_j\|_M^2 = (x_i - x_j)^T M (x_i - x_j), \quad i \neq j \tag{1}$$

where x_i and x_j are two 512-dimensional feature vectors obtained from the last layer of the two sub-CNNs; and M is a positive semidefinite matrix.

Before introducing the proposed loss function with the temporally constrained multi-task learning mechanism, we first present the loss function with common metric learning. Given training samples, we aim to minimize the following loss:

$$\min_M \frac{\lambda}{2}\|M - I\|_F^2 + C\sum_{i,j}\max(0, b - l_{i,j}[1 - \|x_i - x_j\|_M^2])$$
$$s.t. \quad M \succeq 0, i \neq j \tag{2}$$

where λ is a regularization parameter; $\|\cdot\|_F$ denotes the Frobenius norm of a matrix; C is the weight parameter of the empirical loss; b is a constant value satisfying $0 \leq b \leq 1$, which represents the decision margin; $l_{i,j}$ is a label that equals to 1 when x_i and x_j are of the same object and -1 otherwise; and $M \succeq 0$ means that M is a positive semidefinite matrix.

Nevertheless, object appearance can vary a lot in the entire video sequence. It is undesirable to use the same metric to estimate the tracklet affinities over the entire video sequence. In this paper, segment-wise metrics are proposed to be learned within each short-time segment, known as local segment. Meanwhile, to capture the common discriminative information shared by all the segments, a multi-task learning mechanism is proposed to be embedded into the loss function for learning the segment-wise and common metrics simultaneously. Moreover, segments in videos are temporal sequences. Temporally close segments should share more information. Hence, we propose a multi-task learning method incorporating temporal constraints for this learning problem:

$$\min_{M_0,\ldots,M_n}\Big(\frac{\lambda_0}{2}\|M_0 - I\|_F^2 + \sum_{t=2}^{n}\frac{\eta}{2}\|M_t - M_{t-1}\|_F^2 + \sum_{t=1}^{n}[\frac{\lambda}{2}\|M_t\|_F^2 + C\sum_{i,j}h(x_i, x_j)]\Big)$$
$$s.t. \quad M_0, M_1, ..., M_n \succeq 0, i \neq j \tag{3}$$

where λ_0 and λ are the regularization parameters of M_t for $t = 0, 1, ..., n$; n is the total number of segments; M_0 is the common metric shared by all the segments; M_t is the segment-wise metric; $\|\cdot\|_F$ denotes the Frobenius norm of a matrix; the second term of this loss function is the temporal constraint term, in which η is a regularization parameter; $h(x_i, x_j)$ is the empirical loss function; and C is the weight parameter of the empirical loss.

The empirical loss function $h(x_i, x_j)$ used in Equation (3) is expressed as:

$$h(x_i, x_j) = \max(0, b - l_{i,j}[1 - \|x_i - x_j\|_{M_{tot}}^2]); \tag{4}$$
$$M_{tot} = M_0 + M_t, \quad i \neq j,$$
$$\|x_i - x_j\|_{M_{tot}}^2 = (x_i - x_j)^T (M_0 + M_t)(x_i - x_j)$$

where b is a constant value, which represents the decision margin; $l_{i,j}$ is a label that equals to 1 when x_i and x_j are of the same object and -1 otherwise; x_i and x_j are two 512-dimensional feature vectors obtained from the last layer of the two sub-CNNs; and M_{tot} is the metric used for estimating the relative distance between a sample pair.

Intuitively, the common metric M_0 represents the shared discriminative information across the entire video sequence and the segment-wise metric $M_{t>0}$ adapt the metric for each local segment. In the proposed objective function, in Equation (3), the second term is the temporal constraint term, which accounts for the observation that the neighboring segments sharing more information than the non-neighboring segments (see Figure 3 for an illustration). In the implementation, we use the previous segment-wise metric M_{t-1} in temporal space to initialize the current segment-wise metric

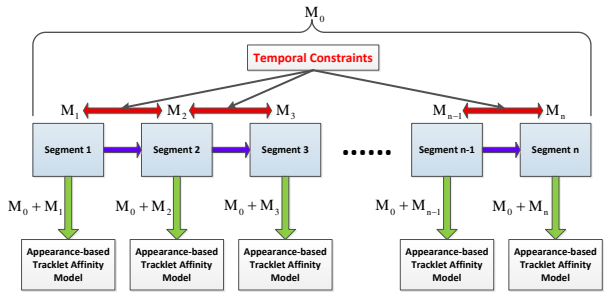

Figure 3. An illustration of the temporally constrained multi-task learning mechanism. n is the total number of the segments and the segments are shown in the temporal space.

M_t, due to the assumption that the neighboring segment-wise metrics are more correlated than the non-neighboring ones.

To learn the parameters of the unified deep model, back-propagation (BP) [25] is utilized. The forward propagation function to calculate the loss of the training pairs is presented in Equation (3). By differentiating the loss function with respect to the two input samples, we have the gradients. The total gradient for back-propagation is the sum of the contributions from the two samples, which is as follows:

$$\nabla G_{total} = 2Cl_{i,j}(M_{tot} + {M_{tot}}^T)(x_i - x_j)\left(\mathbb{I}\{g(x_i, x_j) > 0\}\right) \tag{5}$$

where

$$M_{tot} = M_0 + M_t, \tag{6}$$

$$g(x_i, x_j) = b - l_{i,j}[1 - {\|x_i - x_j\|_{M_{tot}}}^2], \tag{7}$$

$${\|x_i - x_j\|_{M_{tot}}}^2 = (x_i - x_j)^T(M_0 + M_t)(x_i - x_j) \tag{8}$$

and $\mathbb{I}\{\cdot\}$ is the indicator function.

Based on Equations (3) and (5), we can learn the parameters of the unified deep model by stochastic gradient descent via back-propagation. Moreover, the temporally constrained metrics for tracklet affinity models are obtained simultaneously by batch mode stochastic gradient descent.

Online training sample collection is an important issue in the learning of the unified deep model. We take the assumptions similar to those as in [21]: (1) detection responses in one tracklet are from the same object; (2) any detection responses in two different tracklets which have overlaps over time are from different objects. The first one is based on the assumption that the tracklets generated by the dual-threshold strategy are reliable; the second one is based on the fact that one target cannot appear at two or more different locations at the same time, known as spatio-temporal conflict. For each tracklet, κ strongest detection responses are selected as training samples ($\kappa = 4$ in our implementation). Then we use two arbitrarily selected different detection responses from the κ strongest responses of T_i as positive training samples, and two detection responses from the κ strongest responses of two spatio-temporal conflicted tracklets as negative training samples.

Algorithm 1 Online Learning Algorithm for Temporally Constrained Metric Learning

Input:
Feature vectors of online collected training samples $\{{x_i}^t\}$; $i = 1, ..., n_t$, n_t is the number of the samples within segment t; $t = 1, ..., n$, n is the total number of the segments; and learning rate β.

Output:
The learned metrics: $M_0, M_1, ..., M_n$.

Initialize $M_0 = I$ (identity matrix).
for $t = 1, ..., n$ **do**
  if $t == 1$ **then**
    Initialize $M_t = 0$.
  else
    Initialize $M_t = M_{t-1}$.
  end if
  Randomly generate the training pairs $\{x_i, x_j, l_{i,j}\}$ from $\{{x_i}^t\}$. $l_{i,j} = 1$, if x_i and x_j are from one tracklet; $l_{i,j} = -1$, if x_i and x_j are from two different tracklets which have overlaps over time. A total of $2m$ training pairs in an random order are generated, which includes m positive and m negative pairs.
  for $p = 1, ..., 2m$ **do**
    if $l_{i,j}[1 - (x_i - x_j)^T(M_0 + M_t)(x_i - x_j)] > b$ **then**
      M_0=M_0; M_t=M_t.
    else if $l_{i,j} < 0$ **then**
      Compute M_0 and M_t by Equations (9) and (10).
    else
      $M_0 = \pi_{S+}(M_0 - \beta\frac{\partial L}{\partial M_0})$;
      $M_t = \pi_{S+}(M_t - \beta\frac{\partial L}{\partial M_t})$;
      where $\pi_{S+}(A)$ projects matrix A into the positive semidefinite cone.
    end if
  end for
end for

Finally, the common metric M_0 and the segment-wise metrics $M_{t>0}$ are obtained simultaneously through a gradient descent rule. The online learning algorithm is summarized in Algorithm 1.

$$M_0 = M_0 - \beta\frac{\partial L}{\partial M_0} \tag{9}$$

$$M_t = M_t - \beta\frac{\partial L}{\partial M_t} \tag{10}$$

where β is the learning rate.

Meanwhile, the siamese CNN is online fine-tuned through back propagating the gradients calculated by Equation (5).

3. Tracklet Association Framework

In this section, we present the tracklet association framework, in which, we incorporate the temporally constrained metrics learned by the unified deep model to obtain robust appearance-based tracklet affinity models.

3.1. Tracklet Association with Generalized Linear Assignment

To avoid learning tracklet starting and termination probabilities, we formulate the tracklet association problem as a Generalized Linear Assignment (GLA) [34], which does not need the source and sink nodes as in conventional network flow optimization [46, 32, 8, 37]. Given N tracklets $\{T_1, ..., T_N\}$, the Generalized Linear Assignment (GLA) problem is formulated as:

$$\max_X \sum_{i=1}^{N}\sum_{j=1}^{N} P(T_i, T_j)X_{ij} \tag{11}$$
$$s.t. \quad \sum_{i=1}^{N} X_{ij} \leq 1; \sum_{j=1}^{N} X_{ij} \leq 1; X_{ij} \in \{0, 1\}$$

where $P(T_i, T_j)$ is the linking probability between T_i and T_j. The variable X_{ij} denotes that T_i is the predecessor of T_j in temporal domain when $X_{ij} = 1$ and that, they may be merged during the optimization.

3.2. Tracklet Affinity Measurement

To solve the Generalized Linear Assignment (GLA) problem in Equation (11), we need to estimate the tracklet affinity score, or equivalently, the linking probability, $P(T_i, T_j)$, between two tracklets. The linking probability $P(T_i, T_j)$ is defined based on two cues: motion and appearance.

$$P(T_i, T_j) = P_m(T_i, T_j)P_a(T_i, T_j) \tag{12}$$

The motion-based tracklet affinity model $P_m(T_i, T_j)$ is defined as:

$$\begin{aligned} P_m(T_i, T_j) =& \mathcal{N}(p_i^{tail} + v_i^F \Delta t; p_j^{head}, \Sigma) \cdot \\ & \mathcal{N}(p_j^{head} + v_j^B \Delta t; p_i^{tail}, \Sigma) \end{aligned} \tag{13}$$

where p_i^{tail} is the position of the tail response in T_i; p_j^{head} is the position of the head response in T_j; v_i^F is the forward velocity of T_i; v_j^B is the backward velocity of T_j; and Δt is the time gap between the tail response of T_i and the head response of T_j.

In Equation (13), the forward velocity v_i^F is estimated from the head to the tail of T_i, while the backward velocity v_j^B is estimated from the tail to the head of T_j. It is assumed that the difference of the predicted position and the refined position follows a Gaussian distribution.

To estimate the appearance-based tracklet affinity scores, we need to construct the probe set, consisting of the strongest detection response in each tracklet. The probe set is defined as $G = \{g_i\}$, $i = 1, ..., N_s$, in which N_s is the number of tracklets in a local segment. Each T_i has only one selected g_i in G to represent itself.

The appearance-based tracklet affinity model $P_a(T_i, T_j)$ is defined based on the learned temporally constrained metrics:

$$\begin{aligned} & d_{ij}^k = (x_i^k - g_j)^T (M_0 + M_t)(x_i^k - g_j); \\ & d_{ji}^{k'} = (x_j^{k'} - g_i)^T (M_0 + M_t)(x_j^{k'} - g_i); \\ & norm_i^k = \sqrt{(\sum_{j=1}^{N_s} d_{ij}^k)}; \; norm_j^{k'} = \sqrt{(\sum_{i=1}^{N_s} d_{ji}^{k'})}; \\ & d_{ij} = \Big[\sum_k (\frac{d_{ij}^k}{norm_i^k})\Big]/m_i; \; d_{ji} = \Big[\sum_{k'} \frac{d_{ji}^{k'}}{norm_j^{k'}}\Big]/m_j; \\ & P_a(T_i, T_j) = (d_{ij} d_{ji})^{-1} \end{aligned} \tag{14}$$

where x_i^k denotes the feature vector of the kth detection response in T_i; $x_j^{k'}$ denotes the feature vector of the k'th detection response in T_j; $g_i, g_j \in G$; m_i and m_j are the numbers of detection responses of T_i and T_j respectively.

Through Equation (12), we can obtain the predecessor-successor matrix P for the objective function (11). To achieve fast and accurate convergence, P is normalized by column and a threshold ω is introduced to ensure that a reliable tracklet association pair has a high affinity score.

$$P(T_i, T_j) = \begin{cases} P_m(T_i, T_j)P_a(T_i, T_j), \\ \quad if \; P_m(T_i, T_j)P_a(T_i, T_j) \geq \omega \\ \\ 0, \; \text{otherwise} \end{cases} \tag{15}$$

The Generalized Linear Assignment problem in Equation (11) can be solved by the softassign algorithm [17]. Due to missed detections, there may exist some gaps between adjacent tracklets in each trajectory after tracklet association. Therefore, the final tracking results are obtained through a trajectory interpolation process over gaps based on a linear motion model.

4. Experiments

4.1. Datasets

To evaluate the multi-object tracking performance of the proposed method, experiments are conducted on five publicly available datasets: PETS 2009 [15], Town Centre [5], Parking Lot [35], ETH Mobile scene [13] and MOTChallenge [23]. Moreover, a new dataset containing 40 diverse fully annotated sequences is used to evaluate the proposed method. For this new dataset, 10 sequences, which contain 5,080 fames and 52,833 annotated boxes, are used for training; 30 sequences, which contain 19,802 frames and 193,497 annotated boxes, are used for testing.

4.2. Experimental Settings

For the first four datasets evaluation, the proposed siamese CNN is first pre-trained on the JELMOLI dataset

[12] with the loss function in Equation (2). For the MOTChallenge [23], the siamese CNN is first pre-trained on the training set of [23]. For the new dataset, the siamese CNN is first pre-trained on the 10 training sequences. For the regularization parameters in the loss function (3), we set $\lambda_0 = 0.01$, $\lambda = 0.02$ and $\eta = 0.02$. The weight parameter of the empirical loss is set to $C = 0.001$. The learning rate β is fixed as 0.01 for all the sequences. The variance Σ in the motion-based tracklet affinity model in Equation (13) is fixed at $\Sigma = diag[625\ \ 3600]$. A threshold value ω between 0.5 and 0.6 in Equation (15) works well for all the datasets. Moreover, a segment of 50 to 80 frames works well for all the sequences.

For fair comparison, the same input detections and groundtruth annotations are utilized for all the trackers in each sequence. Some of the tracking results are directly taken from the corresponding published papers. For the new dataset, we use DPM detector [14] to generate the detections. The DPM detections with a score above the threshold value -0.3 serve as inputs for all the evaluated trackers in the new dataset.

4.3. Performance Evaluation

Evaluation metrics: We use the popular evaluation metrics defined in [27], as well as the CLEAR MOT metrics [7]: MOTA (↑), MOTP (↑), Recall (↑), Precision (↑), False Alarms per Frame (FAF ↓), False Positives (FP ↓), False Negatives (FN ↓), the number of Ground Truth trajectories (GT), Mostly Tracked (MT ↑), Partially Tracked (PT), Mostly Lost (ML ↓), the number of Track Fragments (Frag ↓) and Identity Switches (IDS ↓). Here, ↑ denotes higher scores indicate better performance, and ↓ denotes lower scores indicate better performance.

Evaluation: To show the effectiveness of joint learning and temporally constrained metrics, two baselines are designed. For **Baseline 1**, the siamese CNN and the metrics are learned separately. We first learn the siamese CNN alone by using the loss function (2), in which the M is fixed as $M = I$. Then the common metric M is learned separately with the features obtained from the previous learned siamese CNN. No segment-wise metrics M_t are learned for Baseline 1. For **Baseline 2**, the unified deep model without the temporally constrained multi-task mechanism is learned for tracklet affinity model. In Baseline 2, we use the loss function in Equation (2) instead of Equation (3) to learn the unified deep model. Moreover, to show the effectiveness of the CNN fine-tuning and the common metric M_0, two more baselines are designed. For **Baseline 3**, the siamese CNN is pre-trained on JELMOLI dataset but without fine-tuning on target dataset using (3). The temporally constrained metrics and the siamese CNN are learned separately. For **Baseline 4**, no common metric M_0 is used. The objective function (3) without the first term is used for this baseline. Note that the siamese CNNs of all the baselines are pre-trained on JELMOLI dataset [12].

From Table 1, 2, 3 and 4, it is found that Baseline 2 achieves overall better performance than Baseline 1 on the evaluated datasets, which proves the effectiveness of the joint learning. Moreover, our method achieves significant improvement in performance on the evaluated datasets, compared with Baseline 2, which validates the superiority of our unified deep model with the temporally constrained multi-task learning mechanism. Our method also achieves overall better performance than Baseline 3 and Baseline 4, which demonstrates the effectiveness of fine-tuning and adding the common metric M_0.

We further evaluate our method on the recent MOTChallenge 2D Benchmark [23]. The qualitative results of our method (CNNTCM) are available on the MOTChallenge Benchmark website [1]. From Table 5, it is found that our method achieves better performance on all evaluation measures compared with a recent work [11] which is also based on the GLA framework. Compared with other state-of-the-art methods, our method achieves better or comparable performance on all the evaluation measures.

Moreover, to further show the generality and effectiveness of the proposed method on large-scale sequences, a new dataset with 40 diverse sequences is built for performance evaluation. 5 state-of-the-art tracking methods with released source codes are used in evaluation for the new dataset. The parameters of each evaluated tracking method are fine-tuned on the 10 training sequences. As shown in Table 6, our method achieves the best performance on MOTA and IDS, which are the most two direct measures for tracklet association evaluation, among all the evaluated methods.

Computation speed: Our system was implemented using the MatConvNet toolbox [36] on a server with a 2.60GHz CPU and a Tesla K20c GPU. The computation speed is subject to the number of targets in a video sequence. The speeds of our method are about 0.38, 0.81, 0.50, 0.60, 0.59, 0.55 (sec/frame) for PETS 2009, Town Centre, ParkingLot, ETH, MOTChallenge, and the new dataset, respectively, excluding the detection step. Note that speed-up can be achieved by further optimization of the codes.

5. Conclusion

In this paper, a novel unified deep model for tracklet association is presented. This deep model can jointly learn the siamese CNN and temporally constrained metrics for tracklet affinity models. The experimental results of Baseline 1 and Baseline 2 validate the effectiveness of the joint learning and the temporally constrained multi-task learning mechanism of the proposed unified deep model. Baseline 3 and Baseline 4 demonstrate the effectiveness of fine-tuning and adding the common metric. Moreover, a new large-

Method	MOTA	MOTP	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Milan *et al.* [29]	90.6%	80.2%	92.4%	98.4%	0.07	59	302	23	91.3%	4.4%	4.3%	6	11
Berclaz *et al.* [6]	80.3%	72.0%	83.8%	96.3%	0.16	126	641	23	73.9%	17.4%	8.7%	22	13
Andriyenko *et al.* [2]	86.3%	78.7%	89.5%	97.6%	0.11	88	417	23	78.3%	17.4%	4.3%	21	38
Andriyenko *et al.* [3]	88.3%	79.6%	90.0%	98.7%	0.06	47	396	23	82.6%	17.4%	0.0%	14	18
Pirsiavash *et al.* [32]	77.4%	74.3%	81.2%	97.2%	0.12	93	742	23	60.9%	34.8%	4.3%	62	57
Wen *et al.* [41]	92.7%	72.9%	94.4%	98.4%	0.08	62	222	23	95.7%	0.0%	4.3%	10	5
Chari *et al.* [9]	85.5%	76.2%	92.4%	94.3%	-	262	354	19	94.7%	5.3%	0.0%	74	56
Baseline1	93.6%	86.3%	96.3%	97.7%	0.13	106	170	19	94.7%	5.3%	0.0%	18	18
Baseline2	94.3%	86.4%	96.6%	97.9%	0.12	94	157	19	94.7%	5.3%	0.0%	16	11
Baseline3	94.0%	86.3%	96.5%	97.8%	0.13	100	163	19	94.7%	5.3%	0.0%	20	12
Baseline4	94.7%	86.4%	97.1%	97.8%	0.13	103	134	19	94.7%	5.3%	0.0%	13	8
Ours	95.8%	86.4%	97.5%	98.4%	0.09	74	115	19	94.7%	5.3%	0.0%	8	4

Table 1. Comparison of tracking results between state-of-the-art methods and ours on PETS 2009 dataset.

Method	MOTA	MOTP	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Leal-Taixe *et al.* [24]	71.3%	71.8%	-	-	-	-	-	231	58.6%	34.4%	7.0%	363	165
Zhang *et al.* [46]	69.1%	72.0%	-	-	-	-	-	231	53.0%	37.7	9.3%	440	243
Benfold *et al.* [5]	64.3%	80.2%	-	-	-	-	-	231	67.4%	26.1%	6.5%	343	222
Pellegrini *et al.* [31]	65.5%	71.8%	-	-	-	-	-	231	59.1%	33.9%	7.0%	499	288
Wu *et al.* [42]	69.5%	68.7%	-	-	-	-	-	231	64.7%	27.4%	7.9%	453	209
Yamaguchi *et al.* [43]	66.6%	71.7%	-	-	-	-	-	231	58.1%	35.4%	6.5%	492	302
Possegger *et al.* [33]	70.7%	68.6%	-	-	-	-	-	231	56.3%	36.3%	7.4%	321	157
Baseline1	54.8%	72.5%	71.1%	85.0%	1.99	895	2068	231	58.0%	31.2%	10.8%	360	268
Baseline2	58.4%	73.0%	72.2%	87.5%	1.63	735	1983	231	59.7%	30.3%	10.0%	325	251
Baseline3	57.1%	72.8%	72.3%	86.5%	1.79	806	1979	231	58.9%	30.7%	10.4%	326	265
Baseline4	63.8%	74.0%	73.2%	90.8%	1.18	530	1915	231	62.8%	29%	8.2%	223	153
Ours	67.2%	74.5%	75.2%	92.6%	0.95	428	1770	231	65.8%	27.7%	6.5%	173	146

Table 2. Comparison of tracking results between state-of-the-art methods and ours on Town Centre dataset.

scale dataset with 40 fully annotated sequences is created to facilitate multi-target tracking evaluation. Furthermore, extensive experimental results on five public datasets and the new large-scale dataset compared with state-of-the-art methods also demonstrate the superiority of our method.

References

[1] Multiple object tracking benchmark. http://motchallenge.net/. 6

[2] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In CVPR, 2011. 7, 8

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In CVPR, 2012. 7, 8

[4] S. H. Bae and K. J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In CVPR, 2014. 1, 8

[5] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In CVPR, 2011. 5, 7

[6] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(9):1806–1819, 2011. 7

[7] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. EURASIP J. Image and Video Processing, 2008. 6

[8] A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In CVPR, 2013. 5

[9] V. Chari, S. L. Julien, I. Laptev, and J. Sivic. On pairwise costs for network flow multi-object tracking. In CVPR, 2015. 7

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005. 1

[11] C. Dicle, O. Camps, and M. Sznaier. The way they move: tracking multiple targets with similar appearance. In ICCV, 2013. 1, 6, 8

[12] A. Ess, B. Leibe, and L. van Gool. Depth and appearance for mobile scene analysis. In ICCV, 2007. 6

[13] A. Ess, K. S. B. Leibe, and L. van Gool. Robust multiperson tracking from a mobile platform. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(10):1831–1846, 2009. 5

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010. 1, 2, 6

[15] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In Winter-PETS, 2009. 5

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 1

[17] S. Gold and A. Rangarajan. Softmax to softassign: Neural network algorithms for combinatorial optimization. Journal of Artificial Neural Nets, 2(4):381–399, 1995. 2, 5

[18] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. arXiv preprint, arXiv:1502.06796, 2015. 2

[19] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In ECCV, 2008. 2

[20] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012. 1, 3

[21] C. H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In CVPR, 2011. 1, 4, 8

[22] L. Leal-Taixe, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In CVPR, 2014. 8

[23] L. Leal-Taixe, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint, arXiv: 1504.01942, 2015. 5, 6

[24] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In ICCV Workshops, 2011. 7

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998. 4

[26] H. Li, Y. Li, and F. Porikli. Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. In BMVC, 2014. 2

[27] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In CVPR, 2009. 6

[28] N. McLaughlin, J. M. D. Rincon, and P. Miller. Enhancing linear programming with motion modeling for multi-target tracking. In WACV, 2015. 8

Method	MOTA	MOTP	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Shu *et al.* [35]	74.1%	79.3%	81.7%	91.3%	-	-	-	14	-	-	-	-	-
Andriyenko *et al.* [2]	60.0%	70.7%	69.3%	91.3%	0.65	162	756	14	21.4%	71.5%	7.1%	97	68
Andriyenko *et al.* [3]	73.1%	76.5%	86.8%	89.4%	1.01	253	326	14	78.6%	21.4%	0.0%	70	83
Pirsiavash *et al.* [32]	65.7%	75.3%	69.4%	97.8%	0.16	39	754	14	7.1%	85.8%	7.1%	60	52
Wen *et al.* [41]	88.4%	81.9%	90.8%	98.3%	0.16	39	227	14	78.6%	21.4%	0.0%	23	21
Baseline1	76.5%	72.8%	86.0%	91.6%	0.78	195	344	14	71.4%	28.6%	0.0%	95	39
Baseline2	80.7%	72.6%	89.5%	92.3%	0.74	185	258	14	78.6%	21.4%	0.0%	63	33
Baseline3	79.7%	72.7%	89.1%	91.8%	0.78	196	269	14	78.6%	21.4%	0.0%	70	34
Baseline4	81.9%	72.7%	89.7%	93.1%	0.65	163	254	14	78.6%	21.4%	0.0%	59	30
Ours	85.7%	72.9%	92.4%	93.5%	0.63	158	188	14	78.6%	21.4%	0.0%	49	6

Table 3. Comparison of tracking results between state-of-the-art methods and ours on ParkingLot dataset.

Method	MOTA	MOTP	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Kuo *et al.* [21]	-	-	76.8%	86.6%	0.891	-	-	125	58.4%	33.6%	8.0%	23	11
Yang *et al.* [44]	-	-	79.0%	90.4%	0.637	-	-	125	68.0%	24.8%	7.2%	19	11
Milan *et al.* [30]	-	-	77.3%	87.2%	-	-	-	125	66.4%	25.4%	8.2%	69	57
Leal-Taixe *et al.* [22]	-	-	83.8%	79.7%	-	-	-	125	72.0%	23.3%	4.7%	85	71
Bae *et al.* [4]	72.03%	64.01%	-	-	-	-	-	126	73.8%	23.8%	2.4%	38	18
Baseline1	68.6%	76.8%	76.7%	90.7%	0.60	812	2406	125	56.8%	32.8%	10.4%	133	31
Baseline2	71.2%	77.0%	78.4%	91.8%	0.54	728	2236	125	60.8%	29.6%	9.6%	75	20
Baseline3	69.6%	76.9%	77.2%	91.3%	0.56	764	2358	125	58.4%	31.2%	10.4%	115	28
Baseline4	72.6%	77.2%	79.1%	92.6%	0.48	653	2161	125	62.4%	28.8%	8.8%	52	18
Ours	75.4%	77.5%	80.2%	94.5%	0.36	486	2050	125	68.8%	24.8%	6.4%	36	6

Table 4. Comparison of tracking results between state-of-the-art methods and ours on ETH dataset.

Method	MOTA	MOTP	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
DP_NMS [32]	14.5%	70.8%	2.3	13,171	34,814	721	6.0%	53.2%	40.8%	3090	4537
SMOT [11]	18.2%	71.2%	1.5	8,780	40,310	721	2.8%	42.4%	54.8%	2132	1148
CEM [29]	19.3%	70.7%	2.5	14,180	34,591	721	8.5%	45%	46.5%	1023	813
TC_ODAL [4]	15.1%	70.5%	2.2	12,970	38,538	721	3.2%	41%	55.8%	1716	637
ELP [28]	25.0%	71.2%	1.3	7,345	37,344	721	7.5%	48.7%	43.8%	1804	1396
CNNTCM (Ours)	29.6%	71.8%	1.3	7,786	34,733	721	11.2%	44.8%	44.0%	943	712

Table 5. Comparison of tracking results between state-of-the-art methods and ours on MOTChallenge Benchmark.

Method	MOTA	MOTP	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
DP_NMS [32]	29.7%	75.2%	31.4%	97.2%	0.09	1,742	130,338	1051	6.7%	37.4%	55.9%	1,738	1,359
SMOT [11]	33.8%	73.9%	43.5%	85.3%	0.72	14,248	107,310	1051	6.2%	58.1%	35.7%	5,507	4,214
CEM [29]	32.0%	74.2%	39.7%	85.1%	0.67	13,185	114,576	1051	11.1%	42.4%	46.5%	2,016	1,367
TC_ODAL [4]	32.8%	73.2%	56.2%	71.3%	2.17	42,913	83,206	1051	21.9%	48.1%	30.0%	3,655	1,577
ELP [28]	34.1%	75.9%	40.2%	89.2%	0.47	9,237	113,590	1051	9.4%	43.9%	46.7%	2,451	2,301
CNNTCM (Ours)	39.0%	74.5%	41.7%	95.1%	0.20	4,058	110,635	1051	11.9%	43.2%	44.9%	1,946	1,236

Table 6. Comparison of tracking results between state-of-the-art methods and ours on the new dataset.

[29] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multi-target tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(1):58–72, 2014. 7, 8

[30] A. Milan, K. Schindler, and S. Roth. Detection- and trajectory-level exclusion in multiple object tracking. In CVPR, 2013. 8

[31] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool. You'll never walk alone: modeling social behavior for multi-target tracking. In ICCV, 2009. 7

[32] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In CVPR, 2011. 5, 7, 8

[33] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In CVPR, 2014. 7

[34] D. Shmoys and E. Tardos. An approximation algorithm for the generalized assignment problem. Mathematical Programming, 62(1):461–474, 1993. 5

[35] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In CVPR, 2012. 5, 8

[36] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. arXiv preprint, arXiv:1412.4564, 2014. 6

[37] B. Wang, G. Wang, K. L. Chan, and L. Wang. Tracklet association with online target-specific metric learning. In CVPR, 2014. 1, 5

[38] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang. Video tracking using learned hierarchical features. IEEE Transactions on Image Processing, 24(4):1424–1435, 2015. 2

[39] N. Wang and D. Y. Yeung. Learning a deep compact image representation for visual tracking. In NIPS, 2013. 2

[40] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In ICCV, 2009. 1

[41] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In CVPR, 2014. 1, 7, 8

[42] Z. Wu, J. Zhang, and M. Betke. Online motion agreement tracking. In BMVC, 2013. 7

[43] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In CVPR, 2011. 7

[44] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In CVPR, 2012. 1, 8

[45] K. Zhang, Q. Liu, Y. Wu, and M. H. Yang. Robust tracking via convolutional networks without learning. arXiv preprint, arXiv:1501.04505, 2015. 2

[46] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In CVPR, 2008. 5, 7