# Multiple Object Tracking by Learning Feature Representation and Distance Metric Jointly

Jun Xiang[1]
junxiang@hust.edu.cn
Guoshuai Zhang[1]
2016110202@scuec.edu.cn
Nong Sang[2]
nsang@hust.edu.cn
Rui Huang[3]
ruihuang@cuhk.edu.cn
Jianhua hou(✉)[1]
zil@scuec.edu.cn

[1] College of Electronics and Information Engineering
South-Central University for Nationalities
Wuhan, China

[2] School of Automation
Huazhong University of Science and Technology
Wuhan, China

[3] School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Guangdong, China

**Abstract**

Designing a robust affinity model is the key issue in multiple object tracking (MOT). With the release of more challenging benchmarks, the association performance based on traditional hand-crafted features is unsatisfactory. Although several current MOT methods adopt deep CNN features directly, they don't resort to metric learning which can significantly improve the model's discriminability. This paper proposes a novel affinity model by learning feature representation and distance metric jointly in a unified deep architecture. Specifically, we design a CNN network to obtain appearance cue tailored towards person Re-ID, and a LSTM network for motion cue to predict target position, respectively. Both cues are then combined with a triplet loss function, which performs end-to-end learning to fuse features in the desired embedding space. Experiments in the challenging MOT benchmark demonstrate that even by a simple Linear Assignment strategy fed with affinity scores, our method achieves competitive results when compared with several state-of-the-art approaches.

## 1 Introduction

Detection based multiple object tracking (MOT) has become increasingly popular thanks to the great improvements in object detection [5, 7, 8, 19]. Given detection responses by a pre-trained detector, this paradigm consists of two key components: an affinity model for estimating the linking probability between detections or tracklets, and an optimization strategy for association. This paper focuses on the first issue.

Feature representation is crucial to affinity model designing. In the last two decades, many efficient feature descriptors have been developed to construct a robust appearance model, including HOG features [5], local binary patterns [28], color histograms [17], etc. Meanwhile, some works have combined other cues like motion or position with the target appearance to further encode the dynamics targets [30, 31]. However, with the release of new standardized benchmarks like MOTChallenge [12, 15], the more challenging scene is presented. For example, the variation between intra-target could be larger than that of extra-target due to great appearance deforming, very low illumination and severe occlusions. In this situation, tracking performance based on the above traditional hand-crafted features is unsatisfactory.

Recently, deep neural networks have been employed in multiple object tracking framework [11, 13, 20, 25]. As a common practice, target representation is learned by convolutional neural networks (CNN) and then is used directly to compute pairwise affinity measure which determines whether two detections belong to the same target. However, this paradigm usually relies on verification loss (i.e. binary classification loss) and seldom considers identity classification loss, which is widely used in the task of person re-identification (Re-ID) and could be exploited here to significantly improve the feature's discriminability.

This paper aims to address two key issues in a unified deep architecture: integrating multiple cues into feature description, and jointly learning feature representation and distance metric. Specifically, we employ a CNN to extract appearance cue tailored towards person Re-ID, and a Long Short-Term Memory (LSTM) network to obtain motion cue aiming at predicting the target position. Finally, the CNN and LSTM are combined with a triplet loss function, which performs end-to-end deep metric learning and generates the embeddings of the fused appearance and motion features. The proposed "CNN+LSTM "model together with a triplet loss function can be considered as learning a mapping function that maps each detection into an embedding space where the difference between detections of the same target is less than that of different targets. Therefore, we can simply compare a pair of detections by computing the Euclidean distance of their embeddings since we directly optimize the network for the task of feature and distance metric learning jointly.Our main contributions are as follows:

- We present a mechanism to combine the strength of CNN and LSTM for multiple cues based feature representation in MOT.
- Within the context of multiple object tracking, we propose a unified deep architecture for associating detections, where feature representation and distance metric are learned jointly.

# 2 Related Work

## 2.1 Appearance model

In tracking-by-detection paradigm, the critical component is to design a robust affinity model to compute the pairwise affinity measure between two detections across time. Most existing methods adopt weak affinity measures based on appearance model such as spatial affinity, e.g. bounding box overlap or Euclidean distance [1, 18], or simple appearance similarity, e.g. intersection kernel with color histogram [32].

A recent trend is to exploit the representation power of deep architecture and develop CNN-based appearance features to model similarity between detections [13, 20, 25, 27]. A common manner in this trend is to train a Siamese CNN to discriminate whether two detections belong to the same target by using a verification loss, i.e.,binary classification. However, models trained with the verification loss only answer the question "How similar are these two detections?" [9]. Intuitively, this answer is "arbitrary" to some degree, since it considers neither "where and when these detections originated", nor other data points except the point pair being compared.

Instead of verification loss, a more appropriate viewpoint is to treat MOT as retrieval or Re-ID problem and to build appearance model based on identity classification loss (identity preserving) . Along this line, Tang et al. [25] developed a Siamese ID-Net for person Re-ID, where the appearance learned by deep networks and body pose information are combined. While identity preserving is considered during training stage in [25], a binary classification (i.e. verification loss) is adopted in the test time due to the unknown number of targets in MOT, thus falling into the verification "trap".

This paper introduces a metric learning strategy, a widely used technique in person Re-ID and bipartite graph matching, to mitigate the above problem. We choose triplet loss function to map CNN features into an embedding space where distances for different appearances of the same target is smaller than that of different targets. The triplet-based feature embedding is more suitable for multiple target tracking since the data association can be viewed as a bipartite graph matching problem.

## 2.2 Motion model

To improve the affinity model's accuracy, motion models are presented and often combined with appearance to predict the target location [3, 16, 20, 27, 29]. Popular motion models used in MOT are often linear [3, 27, 29] with a priori assumption that targets follow a linear movement with constant velocity across frames. Wang et al. [27] construct a Siamese CNN and a traditional linear model to extract appearance and motion features, respectively. The deep appearance features and hand-crafted motion features are then integrated to estimate the linking probability for target association. However, the simple mechanism behind linear model makes it hard to produce a more accurate prediction in a complex and crowded scene, leading to unrealistic or unreasonable trajectories due to the complexity of human motion patterns. To tackle this shortcoming, non-linear motion models are developed to capture more complex dynamics and motion dependency between targets. Milan et al. [16] proposed a RNN-based network to learn complex motion models under the framework of Bayesian filtering, and the temporal dynamics of targets learned by RNN is utilized to perform state prediction and updating as well as track management. Sadeghian et al. [20] presented a LSTM model to predict similar motion patterns by considering the past movements of an object and predicting its future trajectory. To encode long-term temporal dependencies, a hierarchical RNN is used to jointly reason on motion, appearance and interaction cues over a temporal window [20].

Similar to [20], our approach integrates multiple cues into feature description and employs a LSTM network to model motion dependency between targets. However, our system differs in two aspects. First, instead of a regular Siamese CNN in [20], we employ a CNN tailored towards person Re-ID to extract appearance cue. Second, while multiple cues are merged by a RNN with binary classification loss in [20], our approach adopts triplet loss to map the merged features into an embedding space, rendering a mechanism of distance metric learning.

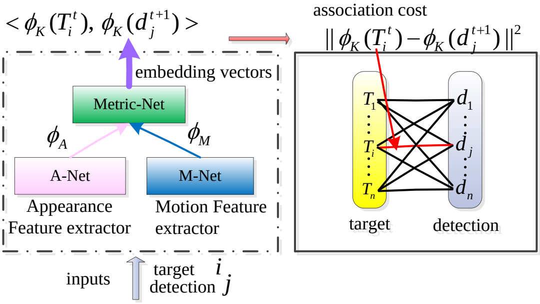


Figure 1: Architecture overview. In affinity model (dashed rectangle), the A-Net and M-Net extract appearance and motion cues, respectively. Both cues are combined in Metric-Net with a triplet loss function to produce the desired embedding features. For association (solid rectangle), a bipartite graph is constructed by association cost between already tracked targets and newly detections, and matching is achieved by Hungarian algorithm.

# 3 Multiple Target Tracking Framework

## 3.1 Architecture Overview

We formulate multiple target tracking as a data association problem. Given a set of already tracked targets at current time $t$, each new detection at a future time, $t+1$, should be associated uniquely with a corresponding target. To this end, an affinity model outputs linking probability to construct a bipartite graph between already tracked targets and newly detections, and Hungarian algorithm is utilized to perform associations across time.

The core task of this paper is to present a novel affinity model by learning feature representation and distance metric jointly in a unified deep architecture. The overview of our framework is shown in Fig. 1. The proposed affinity model consists of three components. The first is a CNN network tailored towards person Re-ID for appearance, denoted as A-Net. The second is a LSTM network used to predict target positions, denoted as M-Net. The output of both networks, denoted by $\phi_A$ and $\phi_M$, are combined in the third part equipped with a triplet loss function, namely Metric-Net. The Metric-Net performs end-to-end learning of multiple cue representations, and produces embedding feature $\phi_K(T_i^t)$ and $\phi_K(d_j^{t+1})$, which are used to output the association cost between $T_i^t$ and $d_j^{t+1}$. In the rest of this section, we describe each component of our method.

## 3.2 Appearance model

The motivation of our appearance model is to develop more discriminative and robust representations for visual feature property. To this end, we employ a CNN with an identity classification loss, which is usually used in person Re-ID.

**Data Collection** Training images are collected from the MOT15 benchmark training set [12] and 5 sequences in the MOT16 benchmark training set [15]. We also collect person identity examples from the CUHK03 and Market-1501 [14] datasets. For validation set, we use the MOT16-02 and MOT16-11 sequences from the MOT16 training set . Overall a total of 2551 identities are used for training and 123 identities for validating.

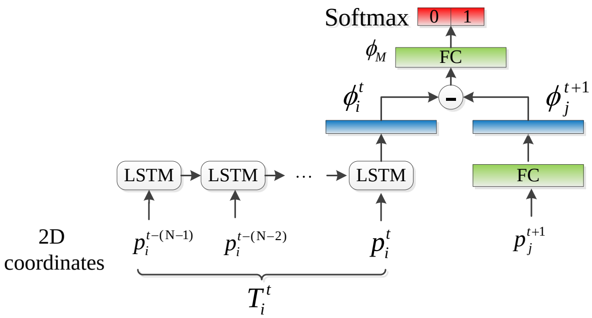


Figure 2: M-Net model. The inputs are 2D coordinates.

**Architecture** We use VGG-16 Net [23] as the base CNN architecture. Specifically, by training VGG-16 to recognize $Y = 2551$ unique identities, the learning can be viewed as a $Y$-way classification problem. Training images are re-sized to $112 \times 224 \times 3$, and each image $I_i$ associates to a ground truth identity label $l_i \in \{1, 2 \cdots Y\}$. The network is trained by the softmax loss to estimate the probability of each image being each label by a forward pass. We also employ bounding box regression loss as in [8] to handle noisy localization of detected targets. During testing, we use the output of fully connected layer $\Phi_{f7}$ as the appearance feature $\phi_A(I_i)$ and refined box localizations as the input for M-Net.

## 3.3 Motion model

Intuitively, the dynamics over trajectories of a specific target encodes its motion pattern that can be exploited to predict future position for the target. In this paper, we explore a Long Short-Term Memory (LSTM) to model target motions over positions. The proposed M-Net model is illustrated in Fig. 2.

**Architecture** The task of motion model is to determine whether a trajectory should be located at a particular position or not. Our LSTM accepts as inputs the positions of trajectory $i$ ( $N$ frames in length), denoted as $T_i^t = [p_i^{t-(N-1)}, \cdots, p_i^{t-1}, p_i^t]$, and produces a $H$-dimensional output $\phi_i^t$ . We also pass a position $p_j^{t+1}$ from trajectory $j$ for time $t+1$ (which we wish to determine whether it corresponds to the true trajectory $T_i^t$ or not) through a fully-connected layer that maps it to a $H$-dimensional vector, denoted as $\phi_j^{t+1}$ . The difference between $\phi_i^t$ and $\phi_j^{t+1}$ is passed to another fully connected layer, followed by a softmax layer to produce a probability estimation over binary classification. For testing, the output of the top FC layer is used as the final motion vector, denoted as $\phi_M(T_i^t, p_j^{t+1})$ .

**Data Collection** We resort to synthetic data augmentation as in [16], by sampling from a simple generative trajectory model learned from MOT15 and MOT16. There are about 100K trajectories in the collected training set, each of 20 frames in length. The data is divided into mini-batches of 10 samples per batch and normalized to the range [-0.5, 0.5], w.r.t. the image dimensions. As mentioned previously, each sample is a data pair consisting of a trajectory $T_i^t$ ($N$ frames in length) and a position $p_j^{t+1}$ . While positive samples are generated by randomly sampling $T_i^t$ and its true position $p_i^{t+1}$ , negative samples consists of $T_i^t$ and a position $p_j^{t+1}$ from a different trajectory $j$.

## 3.4 Metric Learning

In this paper, our metric learning encodes dependencies across appearance and motion automatically by using a triplet loss function denoted as Metric-Net. We first pre-trained appearance and motion model separately. Then we train our Metric-Net with fine-tuning the weights of individual components in an end-to-end fashion by fitting appearance and motion features into the triplet loss function.

**Data Collection** The components of triplet training examples come from image $I_i$ in A-Net, as well as trajectory $T_i^t$ and position $p_j^{t+1}$ in M-Net. Similar to [4, 22], we construct triplet example $Z_i =< Z_i^o,\ Z_i^+,\ Z_i^- >$ with three items. In anchor $Z_i^o = \{I_i, T_i^{t_1}, p_i^{t_1+1}\}$ , both $T_i^{t_1}$ and $p_i^{t_1+1}$ are from a specific target $i$ , and $I_i$ is a randomly sampled detection image from $T_i^{t_1}$ . Anchor-positive $Z_i^+ = \{I_i', T_i^{t_2}, p_i^{t_2+1}\}$ is similar to anchor but with different time stamp $t_2$ . The underlying principle behind our metric learning is to pull together samples from the same class in terms of appearance and motion, while push apart those with either different class in terms of appearance or unreasonable motion state. Consequently, the trajectory and position in Anchor-negative $Z_i^- = \{I_j, T_j^{t_3}, p_k^{t_3+1}\}$ come from a different target $j$ (i.e., $i \neq j$ ) . Note that in this case we don't care about whether $p_k^{t_3+1}$ is the real position of $T_j^{t_3}$ or not. For experiments, we collect triplet examples from MOT15 benchmark training set and 6 sequences of the MOT16 benchmark training set . We use the MOT16-02 sequences from the MOT16 training set as test sets. Overall a total of 851 identities are used for training and 54 identities for testing. We generate the triplets as follows: For each batch of 100 instances, we select 5 persons and generate 20 instances for each person. In each triplet instance, the anchor and anchor-positive are randomly selected from the same identity, and the negative one is also randomly selected, but from the remaining identities.

**Architecture** To train Metric-Net, we design a three-channel CNN-LSTM Model with the shared parameter set. In each channel, one item in triplet training example $Z_i$ is mapped into a learned feature space to form a $4096+H$ dimensional vector by concatenating CNN and LSTM features. A subsequent FC layer is employed for each channel which brings this concatenated feature to a $K = 256$ dimensional embedding space by a triplet loss function, where the embedding feature of $Z_i$ is represented by $\phi_K(Z_i) =< \phi_K(Z_i^o), \phi_K(Z_i^+), \phi_K(Z_i^-) >$ . The learned embedding space has the desirable property that the distance between $\phi_K(Z_i^o)$ and $\phi_K(Z_i^+)$ is less than the distance between $\phi_K(Z_i^o)$ and $\phi_K(\ Z_i^-\ )$ by a predefined margin $\tau$, here $\tau$ is negative.

# 4 Experiments

In this section, we first describe implementation details and evaluation metrics. The performance of each component in our framework as well as the result of ablation study are then analyzed. Finally, we demonstrate validity of the proposed method by comparing with several recent state-of-the-art approaches on the benchmark of MOTChallenge.

## 4.1 Implementation Details

To learn the A-Net, our VGG model is pre-trained on the ImageNet Classification task and fine-turned with the MOT and person identity dataset. The learning rate is set initially to

0.0001 and decreased by 10% every 10 000 iterations. We set the maximum number of iterations to 600 000, which is enough to reach convergence.

We train M-Net from scratch. The weights are initialized from zero-mean Gaussian distributions with the standard deviations 0.01. The bias terms are set to 0. We use RMSprop [26] to minimize the loss. The LSTM is trained with one layer and 300 hidden units, and the iteration step $N$ is experimentally set to 6 for all data sets. The learning rate is set initially to 0.0003 and decreased by 5% every 20 000 iterations. We set the maximum number of iterations to 200 000, which is enough to reach convergence.

Experimentally, the parameter margin of triplet loss function $\tau$ is set to $-2$.

Now we describe data inputting strategy. For target $i$ at time $t$, we pass its final detection image (at time $t$) to A-Net, and pass trajectory $T_i^{t-1} = [p_i^{t-N}, \cdots, p_i^{t-2}, p_i^{t-1}]$ and position $p_i^t$ to M-Net, where the first $N$ entries are inputted into the LSTM and the last entry into the FC layer . The output of our affinity model is the embedding feature, denoted as $\phi_K(T_i^t)$. Similarly, to produce $\phi_K(d_j^{t+1})$, the detection image (at time $t+1$) of $d_j^{t+1}$ is fed into A-Net, and we pass $T_i^t = [p_i^{t-(N-1)}, \cdots, p_i^{t-1}, p_i^t]$ and position $p_j^{t+1}$ of $d_j^{t+1}$ to LSTM.

## 4.2 Evaluation metrics

We follow the standard MOT2D Benchmark challenge [15] for evaluating multiple targets tracking performance. The metrics includes: Multiple Object Tracking Accuracy (MOTA↑), Multiple Object Tracking Precision (MOTP↑), Mostly Track targets (MT↑), Mostly Lost targets (ML↓), False Positives (FP↓), False Negatives (FN↓), Fragmentation (FM↓) and finally ID Switches (IDS↓). For items with (↑), higher scores indicate better results; for those with (↓), lower scores indicate better results.

## 4.3 Experimental Analysis

In this sub-section, we analyze the performance of each component of our model. We conduct experiments on MOT16 to investigate the validity of multiple cues and metric embeddings. 123 person identities collected from MOT16-02 and MOT16-11 are used as test samples. Detections that are considered as true positives for a certain identity are those whose intersection-over-union with the ground truth of the identity are larger than 0.5.

**Validity of A-Net** We evaluate our appearance model for identity verification task. Given the true positive detections for all the test identities, we randomly select 2000 positive pairs assigned to the same identity, and 4000 negative pairs assigned to different identities as our test set. We use the verification accuracy metric, i.e. the ratio of correctly classified pairs. The verification result is obtained by comparing the $L_2$ distance between the extracted features and a threshold. The threshold is obtained on a separate validation dataset by maximizing the verification accuracy, which is set to 0.5 in experiment. We also report the verification result of our A-Net in a Siamese architecture manner (denoted as SA-Net), i.e. an additional FC layer on the top of the twin A-Net is employed to model a 2-way classification.

It can be seen from Table. 1 that our A-Net already produces reasonable verification accuracy. The performance is further improved by SA-Net, from 78.4% to 84.2%. Moreover, we also report Tracking Accuracy (MOTA) of the both networks on MOT-02. While the MOTA result is unsatisfactory due to considering no motion cue, it demonstrates that the A-Net alone can extract meaningful appearance representation for association task. In addition, the result that A-Net achieves a good verification accuracy but a poor tracking accuracy

Table 1: Validity of A-Net

| Model | Verification Accuracy↑ | MOTA↑ |
|---|---|---|
| A-Net | 78.40% | 10.4% |
| SA-Net | 84.20% | 16.2% |

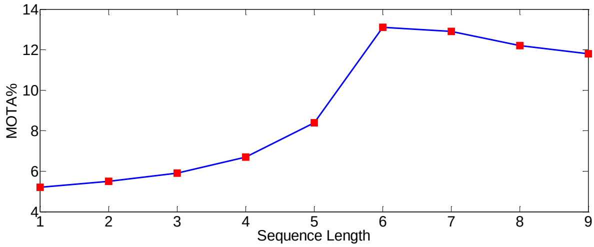


Figure 3: Analysis of the used sequence length for our model on the MOT-02 set

has verifies our previous viewpoint, namely that models trained with the verification loss is "arbitrary" to some degree when applied in assignment task.

**Validity of M-Net** One of the hyper parameters of M-Net is the sequence length $N$ , which is the number of unrolled time steps used while training the LSTM model and enables M-Net being capable to memorize long term dependencies of position cues across time. In this section, we investigate the impact of this parameter. Fig. 3 shows the MOTA score on MOT-02 data set under different sequence length for our LSTM model.

We can see that increasing the sequence length from 1 to 6 positively impacts the MOTA. The performance saturates after 6 frames and then the MOTA slightly decreases. It can be explained by the vanishing gradients. Although the architectures of LSTM provide mechanism to deal with the issue of gradient vanishing to some extent, it does not work well in trajectory modeling problem since long-term occlusions occur frequently in MOT.

**Ablation study** We investigate the contribution of different components of our framework with tracking metrics on MOT-02 dataset. To reveal the contribution of triplet loss, we conduct experiments for the same deep architectures with the verification loss. The evaluation results are presented in Table. 2 and summarized as follow:

- The most important cue is appearance. It can be explained by the fact that representations of people appearance can be learned for varying viewpoint and motion, while less easy to achieve by motion models, especially for monocular video sequences due to the complexity of motion. Note that similar conclusions are also reported in [20, 25].

- The motion cue helps to increase the performance. In highly crowded scenes with clutter and occlusions, our LSTM based motion model can facilitate localization of the targets, while appearance is usually sensitive since the observation likelihood of occluded targets may decrease drastically. In this case, both cues are complementary to make a better performance.

- The triplet loss outperforms the verification loss by a large margin(MOTA from 17% to 23%) on the available datasets. These results echo our claim that using triplet loss to optimize the embedding space is more suitable for retrieval or assignment task.

Table 2: Analysis of our framework using a different set of components. (A) Appearance, (M) Motion, (T) Triplet loss, and (V) Verification loss.

| Tracker | MOTA↑ | MOTP↑ | ML↓ | MT↑ | FP↓ | FN↓ | IDS ↓ |
|---|---|---|---|---|---|---|---|
| A+T | 21.90% | 74.10% | **63.00%** | **11.10%** | 246 | 13642 | 33 |
| M+T | 18.5 % | 74.2 % | 65.00% | 7.40% | 513 | 13944 | 73 |
| A+M+T | **23.00%** | 74.00% | **63.00%** | **11.10%** | 188 | **13542** | **26** |
| A+V | 16.2 % | 74.7% | 67.00% | 5.6% | 247 | 14555 | 148 |
| M+V | 13.10% | **75.20%** | 67.00% | 7.40% | 402 | 14679 | 410 |
| A+M+V | 17.0% | 74.9% | 67.00% | 5.6% | **94** | 14579 | 123 |

Table 3: Results on the MOT16 test dataset. Best in bold, our method is denoted by TripT

| Tracker | MOTA↑ | MOTP↑ | MT↑ | ML↓ | FP↓ | FN↓ | IDS↓ | HZ↑ |
|---|---|---|---|---|---|---|---|---|
| LMP [25] | **48.8** | **79** | **18.20%** | **40.10%** | 6654 | **86245** | 481 | 0.5 |
| AMIR [20] | 47.2 | 75.8 | 14.00% | 41.60% | 2681 | 92856 | 774 | 1 |
| **TripT(Ours)** | 45.6 | 76.0 | 13.00% | 46.90% | **1793** | 96970 | **432** | 0.6 |
| Quad-CNN [24] | 44.1 | 76.4 | 14.60% | 44.90% | 6,388 | 94,775 | 745 | 1.8 |
| oICF [10] | 43.2 | 74.3 | 11.30% | 48.50% | 6,651 | 96,515 | 381 | 0.4 |
| MHT-DAM [11] | 42.9 | 76.6 | 13.60% | 46.90% | 5,668 | 97,919 | 499 | 0.8 |
| LINF1 [6] | 41 | 74.8 | 11.60% | 51.30% | 7,896 | 99,224 | 430 | 1.1 |
| EAMTT-pub [21] | 38.8 | 75.1 | 7.90% | 49.10% | 8,114 | 102,452 | 965 | 11.8 |
| OVBT [2] | 38.4 | 75.4 | 7.50% | 47.30% | 11,517 | 99,463 | 1,321 | 0.3 |

**Comparison with the state of the art** We compare our method, denoted as TripT, with several state of the art approaches on the MOT16 test set. The quantitative results are presented in Table 3, and our method is among the top performing trackers. AMIR [20] is the most relevant approach to ours, where appearance, motion and interaction cues are merged for affinity model and Hungarian algorithm is used for association. We believe our results can be further improved by utilizing other cues such as interaction in AMIR or robust optical flow for motion model. Compared with Quad-CNN [24] which also adopted the metric learning but with the Quadruplet loss, we obtained an improvement in MOTA (45.6% versus 44.1%). In the top one, i.e. LMP [25], more complex graph-cut association strategy is employed in a batch of several frames, and a post trajectory estimation step is added in LMP to handle missing detections. In contrast, we adopt a simple and local optimization strategy (Linear Assignment) to match the tracks with detections at a time frame. We speculate that the ranking first performance of LMP is partly attributed to the more sophisticated optimization strategy as well as post trajectory estimation, and we believe these two factors can be employed in our method to obtain better tracking results. It should be noted that compared with AMIR, LMP and Quad-CNN, our method achieves better IDS. Lower ID number often implies better capability to handle occlusions and more robust tracking performance, and we attribute this desirable property to the proposed framework by learning feature representation and distance metric jointly.

# 5 Conclusion

In this work, we have presented a novel affinity model for data association under a unified deep architecture, where multiple cue feature representation and distance metric are jointly learned in an end-to-end fashion. Experiments in the challenging MOT benchmark show that even employing a simple linear program algorithm for association, the proposed affinity model yields very competitive results compared with the state-of-the-art approaches. In the future, we believe that by merging other cues for feature representation and more effective but often sophisticated optimization strategies for association, better tracking results could be achieved.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No.61671484 , No.61701548, by the Hubei Natural Science Foundation under Grant No.2018CFB503, and by the Fundamental Research Funds for the Central Universities, South-Central University for Nationalities, under Grants No.CZZ18001 and No.CZY18046.

## References

[1] Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1926–1933. IEEE, 2012.

[2] Yutong Ban, Sileye Ba, Xavier Alameda-Pineda, and Radu Horaud. Tracking multiple persons based on a variational bayesian model. In *European Conference on Computer Vision*, pages 52–67. Springer, 2016.

[3] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009.

[4] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Computer Vision and Pattern Recognition*, pages 1335–1344, 2016.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[6] Loïc Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frédéric Lerasle. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In *European Conference on Computer Vision*, pages 774–790. Springer, 2016.

[7] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[8] Ross Girshick. Fast r-cnn. *Computer Science*, 2015.

[9] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[10] Hilke Kieritz, Stefan Becker, Wolfgang Hübner, and Michael Arens. Online multi-person tracking using integral channel features. In *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pages 122–130. IEEE, 2016.

[11] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

[12] 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.

[13] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 33–40, 2016.

[14] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.

[15] 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.

[16] Anton Milan, Seyed Hamid Rezatofighi, Anthony R Dick, Ian D Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, pages 4225–4232, 2017.

[17] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003.

[18] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[20] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 4(5):6, 2017.

[21] Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*, pages 84–99. Springer, 2016.

[22] 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.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5620–5629, 2017.

[25] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017.

[26] T Tieleman and G Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. Technical report, Technical report, 2012.

[27] Bing Wang, Li Wang, Bing Shuai, Zhen Zuo, Ting Liu, Kap Luk Chan, and Gang Wang. Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2016.

[28] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[29] Jun Xiang, Nong Sang, Jianhua Hou, Rui Huang, and Changxin Gao. Hough forest-based association framework with occlusion handling for multi-target tracking. *IEEE Signal Processing Letters*, 23(2):257–261, 2016.

[30] Jun Xiang, Nong Sang, Jianhua Hou, Rui Huang, and Changxin Gao. Multitarget tracking using hough forest random field. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(11):2028–2042, 2016.

[31] Bo Yang and Ramakant Nevatia. Multi-target tracking by online learning a crf model of appearance and motion patterns. *International Journal of Computer Vision*, 107(2): 203–217, 2014.

[32] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*, pages 343–356. Springer, 2012.