

# Improving Online Multiple Object tracking with Deep Metric Learning

Michael Thoreau, Navinda Kottege

**Abstract**—Tracking by detection is a common approach to solving the Multiple Object Tracking problem. In this paper we show how deep metric learning can be used to improve three aspects of tracking by detection. We train a convolutional neural network to learn an embedding function in a Siamese configuration on a large person re-identification dataset offline. It is then used to improve the online performance of tracking while retaining a high frame rate. We use this learned appearance metric to robustly build estimates of pedestrian’s trajectories in the MOT16 dataset. In breaking with the tracking by detection model, we use our appearance metric to propose detections using the predicted state of a tracklet as a prior in the case where the detector fails. This method achieves competitive results in evaluation, especially among online, real-time approaches. We present an ablative study showing the impact of each of the three uses of our deep appearance metric.

## I. INTRODUCTION

Accurately tracking objects of interest such as pedestrians and vehicles in video streams is an extremely important problem with widespread applications in many fields including surveillance, robotics and industrial safety among others. The problem of Multiple Object Tracking (MOT) in video has mostly been addressed in recent literature using the ‘tracking by detection’ framework. In tracking by detection, detections are combined to estimate the trajectories of tracked objects. Solutions can generally be grouped into online and batch processes. The difference being, online solutions use measurements as they arrive while a batch process may build globally optimal trajectories by considering measurements at all times.

In this paper we present an online approach to solving the MOT problem for pedestrian tracking and evaluate it on the MOTChallenge dataset [1], [2].

Motivated by the large amounts of labelled data now available for pedestrian re-identification problems, the proposed method uses a deep-learning approach to appearance modelling. We present a convolutional neural network, trained in a Siamese configuration to produce a discriminative appearance similarity metric for pedestrians.

We present three ways in which this deep appearance metric learning can be used in MOT and propose a method using two of these components to achieve competitive performance. We compare our results to those of other methods and selectively evaluate each use of the proposed appearance metric. First we show how a learned appearance metric can be used to improve the *assignment* of candidate detections to form short tracks (tracklets) as the first step in creating longer

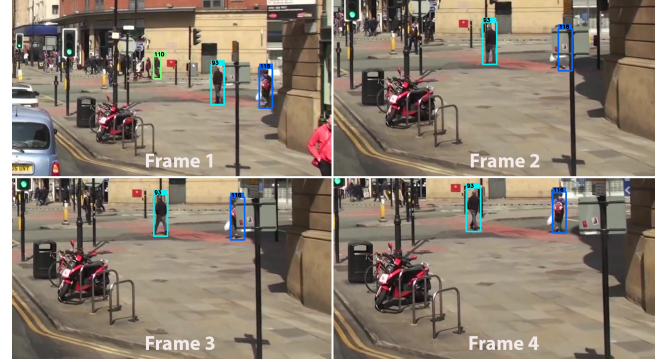


Fig. 1: An example of tracking through occlusion over four consecutive frames using proposed method.

optimal tracks. Next we show how the same metric learner can perform *detection boosting* to reduce false negatives where detections are missing within a person’s track. Lastly the deep appearance metric is used to perform iterative appearance based merging of tracklets to form longer tracks, a process we call *tracklet association*. We accomplish this as an online process, with a playback delay of only a few seconds, at a frame rate suitable for real-time applications.

The rest of the paper is organised as follows; section II describes the related approaches in the literature, section III introduces the proposed Siamese Deep Metric Tracker, section IV evaluates the proposed method against the publicly available Market-1501 dataset of the MOTChallenge, section V discusses the evaluation results and section VI concludes the paper.

## II. RELATED WORK

Solutions to the multiple object tracking problem fall in to two distinct categories; batch and online processing. In batch processing, detections are combined in a global sense, rather than frame by frame, to form optimal tracks [3], [4], [5]. Despite the apparent performance advantages of batch methods as described by Luo et al. in their extensive literature review [6], we consider only online methods in this work, where filtered tracks are available with little to no delay, motivated by potential real-time applications in surveillance, robotics and industrial safety.

In some online approaches, tracklet states are estimated by a probabilistic model such as a Kalman filter [7], [8]. Others have used deep learning to learn to estimate the motion of tracked objects from data, including estimating the birth and death of tracks [9].

A difficult aspect of tracking by detection is solving the data association problem present when grouping detections

<sup>3</sup> M. Thoreau and N. Kottege are with the Robotics and Autonomous Systems Group, CSIRO, Pullenvale, QLD 4069, Australia. All correspondence should be addressed to navinda.kottege@csiro.au

or merging tracklets. Some work is present in the literature that use confidence estimation to aid in the data association problem by prioritising high confidence tracks [10], [11]. Others leverage image information, where even simple appearance modelling has been shown to make data association more robust [12]. Appearance modelling plays a larger role in single object tracking, where only appearance is used to track objects given a prior [13].

More recently, the availability of labelled data has motivated methods utilising deep learning for appearance modelling. Siamese networks have been used for single object tracking to great effect by Feichtenhofer et al. and Liu et al., where the deep appearance model is used to search successive frames [14], [15].

In multiple object tracking, online learning has been used to discriminate between tracked objects based on appearance albeit at limited speed due to computational complexity [10].

Some methods using deep learning have achieved outstanding results on the MOT Challenge [16], [17]. For example, Leal-Taixé et al. achieves good results with a Siamese network, however with a requirement for a gallery of images to be stored from past tracks to do re-identification [18]. Wojke et al. solves this by using a deep similarity metric learning network to store a gallery of metrics [19]. He et al. goes one step further and uses a deep recurrent network to compute an appearance metric which incorporates temporal information from the tracked object, to good effect [20].

Informed by the available literature, we have developed our proposed Siamese Deep Metric Tracker to address the problem of online multiple object tracking, and evaluate how deep appearance modelling may best be used in real-time applications.

### III. SIAMESE DEEP METRIC TRACKER

Here we present our proposed Siamese Deep Metric Tracker to perform online multiple object tracking. A strong appearance model is central to this proposed method. We use a single deep neural network, detailed in section III-B, to enable or assist three components of our object tracking algorithm shown at a high level in figure 2. We solve the problem in multiple stages; firstly, detections are *Assigned*

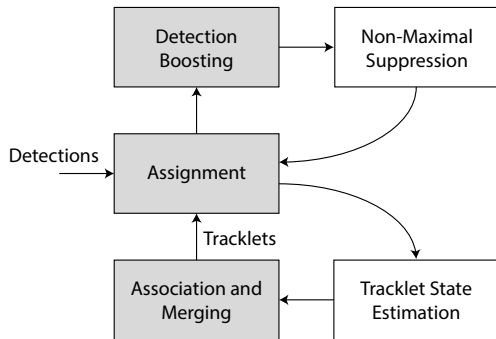


Fig. 2: The proposed process where *Assignment*, *boosting*, and *tracklet association* components benefit from the use of deep appearance modelling.

to tracklets, as detailed in section III-D; detections are then *Boosted* as described in section III-F; and finally tracklets are *Associated* as described in section III-G.

#### A. Notation

We use the following notation in all equations, explanations and algorithm listings in this paper. Let the set of estimated tracklets be  $\mathcal{T}$ , containing  $J$  tracklets  $T_j$ . Let the estimated state of tracklet  $j$  at time  $t$  be  $T_j^t$  and the predicted state of tracklet be  $T_j^{t'}$ . Let a set of detections at time  $t$  be  $\mathcal{D}^t$  containing  $I$  detections  $D_i$ .

#### B. Deep Similarity Metric

A robust appearance model can improve simple object tracking by preventing tracks from drifting to false positive detections, and by enabling objects to be tracked through occlusion.

At each time step we compute a feature vector  $f \in \mathbb{R}^{128}$  for each candidate detection in a single batch. Computing all features in a batch is an efficient use of GPU resources, taking only  $\approx 20$  ms for a typical batch of 40 image patches. The network, with layers listed in table I, uses pre-trained convolutional layers from VGG-16 [21], followed by two fully connected layers with batch and  $l_2$  normalisation on the output layer. The use of pre-trained networks as feature extractors in Siamese/triplet networks has been shown to reduce the number of iterations required for convergence and improve accuracy [22]. Euclidean distance between feature vectors lying within a unit hyper-sphere measures the distance  $d_a = ||f_1 - f_2||$  between two input patches in the appearance similarity space. The appearance affinity  $A_a$  between two patches is  $A_a = 1 - d_a$ . We use an optimal

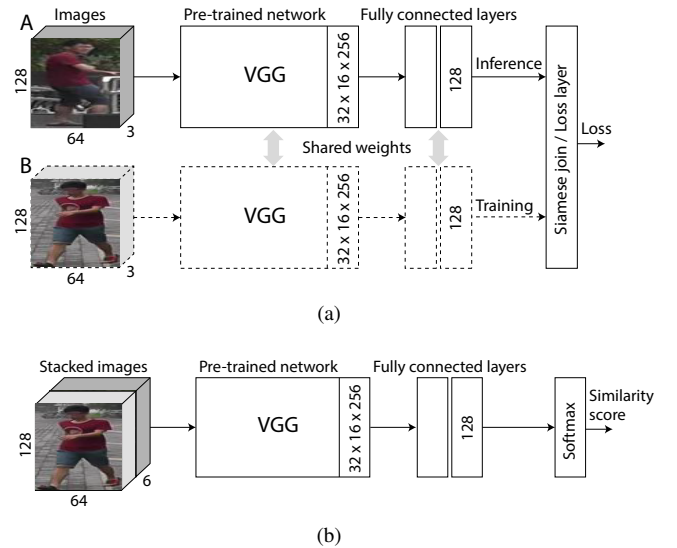


Fig. 3: (a) The proposed Siamese network that learns a similarity metric offline, with a margin contrastive loss, (b) An alternative Siamese network that takes two images as an input and outputs a similarity score.

affinity threshold  $\tau_a = 0.895$ , determined offline, to separate similar and dissimilar pairs.

Two implementations of Siamese networks are shown in figure 3. Figure 3(a) is the proposed implementation, using margin contrastive loss with a fixed margin of 0.2. Figure 3(b) is an alternative implementation, using a learned softmax classifier to give a similarity score between the input images. In our approach, we compute the feature vector for a detection and store it with the state of the tracklet at the time of the detection, meaning that we don't have to store a gallery of images for each tracked object. We assume that the appearance metric computed from the detection will closely match the appearance metric of the true bounding box of the subject. This assumption appears to hold during testing, as bounding boxes are usually well regressed to the true bounding box of the detected object.

### C. Training

The deep similarity network was trained on the Market-1501 pedestrian re-identification dataset [23], containing  $\approx 32,000$  annotated images of 1501 unique pedestrians in six camera views. Triplet loss has recently been used to good effect in training networks for pedestrian re-identification [22]. Networks using triplet loss have been known to be difficult to train, due to a stagnating training loss. Batch-hard example mining has been shown to improve convergence when training with triplet loss [22]. Our approach uses batch-hard sampling to train our network in a Siamese fashion using margin contrastive loss in a large batch. We sample 4 images each from 32 identities, compute their feature vectors in a forward pass and select the hardest pairings, maximising Euclidean distance between feature vectors for positive pairs and minimising distance for negative pairs, for each of the 128 images.

### D. Detection Assignment

Detections are combined across time to estimate the trajectory of a tracked object. This algorithm is shown in listing 1 and detailed below. The motion of small segments, tracklets, are estimated via a Kalman filter with a constant velocity constraint. Tracklet states are predicted at each time step, but are considered inactive after two predictions without being assigned a detection. A tracklet's state is predicted for another 90 steps for tracklet association, discussed in section III-G. The association of new detections to the set of active tracklets is solved as a data association problem using the

---

#### Algorithm 1 Detection Assignment

---

```

1: for  $T_j, D_i^t \in \mathcal{T}, \mathcal{D}^t$  do
2:   if  $A_m(T_j, D_i^t) > \tau_m \wedge A_a(T_j, D_i^t) > \tau_a$  then
3:      $\tilde{A}_{j,i} \leftarrow A(T_j, D_i^t)$ 
4:   else
5:      $\tilde{A}_{j,i} \leftarrow 0$ 
6:   end if
7: end for
8:  $\tilde{M} \leftarrow H(\tilde{A})$  {Hungarian Algorithm Matching}
9: for  $T_j, D_i^t \in \mathcal{T}, \mathcal{D}^t$  do
10:  if  $M_{j,i} = 1$  then
11:    update state of tracklet  $T_j$  with detection  $D_i^t$ 
12:  end if
13: end for

```

---

Hungarian algorithm [24]. The Hungarian algorithm maximises the affinity between tracklets and assigned detections, provided in the affinity matrix  $\tilde{A}$ , and creates entries in the matching matrix  $\tilde{M}$ . The affinity used to assign candidate detections to tracklets is a combination of motion affinity, preferencing detections close to the predicted position of the tracklet, and appearance affinity which attempts to match the tracklet with a detection whose appearance is closest to stored appearance information. Motion affinity is implemented as the Intersection over Union (IoU) [25] between a candidate detection  $D_i^t$  and the predicted bounding box of the tracklet  $T_j^{t'}$ , as shown in equation 1. Motion affinity is constrained to be strictly greater than a motion affinity threshold  $\tau_m = 0.3$  for assignment.

$$A_m(T_j, D_i^t) = IoU(T_j^{t'}, D_i^t) \quad IoU(b_1, b_2) = \frac{b_1 \cap b_2}{b_1 \cup b_2} \quad (1)$$

Appearance affinity is computed as the mean affinity between a candidate detection's feature vector and the stored feature vectors for a tracklet, shown in equation 2 with  $t_0$  denoting the first state of the tracklet. A subset of  $N$  past states of the tracklet is used for computational tractability, in practice  $N \leq 20$ .

$$A_a(T_j, D_i^t) = \frac{1}{N} \sum_{n=n_0}^N f(T_j^n, D_i^t) \quad n \in \{t_0, t-1\} \quad (2)$$

$$f(T_j^n, D_i^t) = 1 - \|f_1 - f_2\| \quad (3)$$

The total affinity, shown in equation 4, is a combination of appearance and motion affinity, balanced by the parameter  $\lambda$ , typically between 0.3 and 0.7. A value of 0 may be used to ignore appearance entirely when computing assignments, potentially improving frame rate under some implementations.

$$A(T_j^t, D_i^t) = \lambda A_a(T_j, D_i^t) + (1 - \lambda) A_m(T_j, D_i^t) \quad (4)$$

TABLE I: Similarity Network Structure

Layer	Output shape
Input	$128 \times 64 \times 3$
VGG-16	$32 \times 16 \times 256$
Fully connected	128
Fully connected	128
Batch normalisation	128
$l_2$ normalisation	128

### E. Tracklet Confidence

A minimum length requirement  $\tau_l = 6$  is imposed on tracklets for them to be considered positive. Tracks containing less than six states are considered negative and therefore are not reported. The mean confidence of detections assigned to a given tracklet is also used to filter out low confidence tracklets, with a minimum mean confidence of  $\tau_c = 0.2$  used in practice. The average cost of assigning detections to a tracklet is used to estimate confidence in it being positive. A tracklet with a high mean assignment cost is likely to be varying in appearance or in motion and is considered negative. Tracklet association and boosting considers only positive tracks to avoid joining false positives with true positives.

### F. Detection Boosting

In the case that in a given frame, there exists no detection which matches to a tracklet, but the tracked object is not occluded or out of frame, we wish to re-identify that person. Using the predicted location of the object as a prior, we perform dense sampling around the prediction and select the candidate bounding box which maximises appearance affinity and satisfies the appearance affinity constraint  $\tau_a = 0.895$ . This detection is added to the detection set and association is performed again, as shown in figure 2. In order to prevent track drift, boosting is limited to no more than once per two frames per track. To stop partial detections from drifting to a true person via boosting and therefore adding false positives, Non Maximum Suppression (NMS) is performed on the detections with a NMS-IoU threshold of 0.5.

### G. Tracklet Association

Targets may be tracked through occlusion by matching tracklets across time using their appearance. Our association algorithm is shown in listing 2 and described below. Due to uncertainties in camera and target motion, a much looser motion constraint is used to associate tracklets, requiring only a small overlap between the predicted bounding box of the older tracklet and the first bounding box of the newer track i.e.  $IoU(T_j^{t'}, T_k^t) > 0$ .

$$A_a(T_j, T_k) = \frac{1}{N} \frac{1}{M} \sum_{n=n_0}^N \sum_{m=m_0}^M f(T_j^n, T_k^m) \quad (5)$$

As tracking is done in the image plane, changes in camera motion may frequently violate the constant velocity constraint imposed by our Kalman filter based tracking. By building small tracklets with a stricter motion constraint and linking high confidence tracklets in to longer tracks with a looser motion constraint, intuitively our tracking may be robust to changes in camera motion. Tracklet association need not run at every time step, once every 20 time steps is sufficient to not impact performance, resulting in a higher refresh rate.

After tracklets have been merged, temporal gaps are filled by interpolation with a constant velocity, giving a reasonable

estimate for the state of the object while it is occluded, this can be seen in figure 1.

## IV. EVALUATION

The Siamese deep metric network was validated on a subset of Market-1501 dataset not used for training. The network achieved an area under the receiver operator characteristic curve of 0.98 after 90,000 training iterations, with an equal mix of positive and negative pairs and distractors sampled from the background.

### A. CLEAR MOT Metrics

The CLEAR MOT [26] metrics are used here to compare our performance to others, as well as compare the benefits of each of the uses of our deep appearance model. The specific metrics we use ( $\uparrow$  denotes metrics in which a higher score is better,  $\downarrow$  denotes metrics in which a lower score is better):

- MOTA  $\uparrow$ , combines FP, FN and IDs to give a single metric to summarise accuracy.
- FP  $\downarrow$ , is the number of false positive bounding boxes.
- FN  $\downarrow$ , is the number of false negative bounding boxes.
- IDs  $\downarrow$ , is the number of times tracked targets swap ID's.
- FPS  $\uparrow$ , is the update frequency, an important metric for real-time applications.

### B. MOT16 results

A selection of methods suitable for real-time applications was made for comparison. Online approaches that achieve an update rate of greater than 10 Hz on the computationally intensive test set using the public detections are shown in table III compared to our approach.

Among the selected approaches, our method achieves a competitive tracking accuracy (MOTA) for a relatively simple method. At only 602, our method achieves the second lowest number of ID switches (IDs), second only to a method with significantly more false negatives.

We performed repeated testing while enabling/disabling certain aspects of our algorithm, presented in table II. The best performing method from this ablative testing was used in testing presented in table III. The best method did not

---

### Algorithm 2 Tracklet Association

---

```

1: for  $T_j \in \mathcal{T}$  do
2:    $C = \emptyset$ 
3:   for  $T_k \in \mathcal{T} - T_j$  do
4:     if confidence constraints met on  $T_j$  and  $T_k$  temporal
       overlap exists then
5:       if  $IoU(T_j^t, T_k^{t'}) > 0$  then
6:          $C \leftarrow C \cup T_k$  {build set of candidate matches}
7:       end if
8:     end if
9:   end for
10:  select best match  $T_o = \arg \max_l A_a(T_j, T_l) \forall T_l \in C$ 
11:  merge tracklets  $T_j \leftarrow T_j \cup T_o$ 
12:  go to 1 until no matches found
13: end for

```

---

TABLE II: Ablative testing performed on the MOT16 training set, with and without appearance modelling for detection assignment, detection boosting, and tracklet association. Best results in each category appear in bold. Tracking metrics are discussed in section IV-A.

Method	MOTA $\uparrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$	FPS $\uparrow$
SDMT ( $\lambda = 0.5$ )	<b>34.6</b>	6,014	65,863	317	29.8
SDMT ( $\lambda = 0$ )	34.3	6,541	65,651	<b>295</b>	29.6
SDMT ( $\lambda = 1$ )	33.9	6,512	66,040	373	28.7
SDMT (w/ boosting)	34.2	6,743	<b>65,533</b>	334	25.9
SDMT (w/o association)	32.4	<b>3,968</b>	69,965	686	31.6
SDMT (w/o appearance modelling)	32.9	4,069	69,468	587	<b>96.8</b>

include boosting and used a lambda value of  $\lambda = 0.5$ . Changing  $\lambda$  to 0 or 1 reduced accuracy on the training set. Adding boosting to the optimal method reduced false negatives but significantly increased false positives. Removing tracklet association, or appearance modelling entirely significantly reduced tracking accuracy. The method without any appearance modelling removed the need to compute feature vectors for each detection, significantly increasing the update rate.

## V. DISCUSSION

We found that using our deep appearance metric for detection assignment and tracklet association improved the performance of multiple object tracking. Detection boosting was found to hurt the accuracy of our tracking, despite reducing the number of false negatives as intended. This was likely due to the high recall rate of 43% but relatively low precision of the DPM v5 detections provided with the test sequences [2]. Boosting is most useful when there exists no detection for a given target, yet the target is not occluded or out of frame. It is possible that this case does not occur often in the MOT16 dataset, limiting effect of reducing false negatives. Tracklets built from false positive detections that contain some part of a true object, may be boosted, causing drift towards the true object. This may lead to the tracks being merged, increasing false positives.

Other similar approaches such as work by Wojke et al. were not tested using the publicly available detections, or did not make their results available online [19], [20].

TABLE III: Results on the MOT16 [2] test set, compared to a selection of algorithms suitable for real-time applications. Best results in each category appear in bold.

Method	MOTA $\uparrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$	FPS $\uparrow$
MOTDT	<b>47.6</b>	9,253	<b>85,431</b>	792	20.6
SAD-T	43.4	15,341	87,086	763	11.4
FullTest	40.7	14,354	92,650	3,864	236.8
<b>SDMT (ours)</b>	39.6	11,130	98,343	602	19.7
DeepAC	38.8	5,444	103,174	2,886	21.1
EAMTT-pub	38.8	8,114	102,452	965	11.8
ERCTracker	32.3	<b>1,193</b>	121,333	953	32.0
cppSORT	31.5	3,048	120,278	1,587	<b>687.1</b>
GMPHD_HDA	30.5	5,169	120,970	<b>539</b>	13.6

## VI. CONCLUSIONS

We presented three uses of deep appearance metric learning for improving multiple object tracking, and demonstrated how two of these uses significantly improved tracking accuracy. Our method achieved competitive results for online methods suitable for real-time applications. Our ablative testing may be used to inform further use of deep appearance metrics in multiple object tracking.

## VII. ACKNOWLEDGEMENTS

The authors would like to thank Benjamin Tam, Lachlan Tychsen-Smith and Nicholas Panitz for their assistance during this work.

## REFERENCES

- [1] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking," *arXiv:1504.01942 [cs]*, Apr. 2015.
- [2] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," Mar. 2016.
- [3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *arXiv:1506.01497 [cs]*, June 2015.
- [4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *arXiv:1512.02325 [cs]*, vol. 9905, pp. 21–37, 2016.
- [5] L. Tychsen-Smith and L. Petersson, "DeNet: Scalable Real-time Object Detection with Directed Sparse Sampling," *arXiv:1703.10295 [cs]*, Mar. 2017.
- [6] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple Object Tracking: A Literature Review," *arXiv:1409.7618 [cs]*, Sept. 2014.
- [7] E. Bochinski, V. Eiselein, and T. Sikora, "High-Speed tracking-by-detection without using image information," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug. 2017, pp. 1–6.
- [8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," *arXiv:1602.00763 [cs]*, pp. 3464–3468, Sept. 2016.
- [9] A. Milan, S. H. Rezaeifighi, A. Dick, I. Reid, and K. Schindler, "Online Multi-Target Tracking Using Recurrent Neural Networks," Apr. 2016.
- [10] S. H. Bae and K. J. Yoon, "Confidence-Based Data Association and Discriminative Deep Appearance Learning for Robust Online Multi-Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.
- [11] —, "Robust Online Multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1218–1225.
- [12] V. Takala and M. Pietikainen, "Multi-object tracking using color, texture and motion," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–7.

- [13] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2544–2550.
- [14] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to Track and Track to Detect," *arXiv:1710.03958 [cs]*, Oct. 2017.
- [15] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese Instance Search for Tracking," May 2016.
- [16] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-End Comparative Attention Networks for Person Re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, July 2017.
- [17] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple Object Tracking with High Performance Detection and Appearance Feature," *arXiv:1610.06136 [cs]*, Oct. 2016.
- [18] L. Leal-Taixé, C. C. Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," Apr. 2016.
- [19] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," Mar. 2017.
- [20] Q. He, J. Wu, G. Yu, and C. Zhang, "SOT for MOT," *arXiv:1712.01059 [cs]*, Dec. 2017.
- [21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Sept. 2014.
- [22] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv:1703.07737 [cs]*, Mar. 2017.
- [23] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable Person Re-identification: A Benchmark," in *IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1116–1124.
- [24] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.
- [25] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An Advanced Object Detection Network," *arXiv:1608.01471 [cs]*, pp. 516–520, 2016.
- [26] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The CLEAR 2006 Evaluation," in *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Apr. 2006, pp. 1–44.