DASOT: A Unified Framework Integrating Data Association and Single Object Tracking for Online Multi-Object Tracking

Qi Chu,[1*] **Wanli Ouyang,**[2] **Bin Liu,**[1] **Feng Zhu,**[3] **Nenghai Yu**[1]
[1]University of Science and Technology of China, China
[2]The University of Sydney, SenseTime Computer Vision Research Group, Australia
[3]SenseTime Group Limited, China
{qchu, flowice, ynh}@ustc.edu.cn, wanli.ouyang@sydney.edu.au, zhufeng@sensetime.com

Abstract

In this paper, we propose an online multi-object tracking (MOT) approach that integrates data association and single object tracking (SOT) with a unified convolutional network (ConvNet), named DASOTNet. The intuition behind integrating data association and SOT is that they can complement each other. Following Siamese network architecture, DASOTNet consists of the shared feature ConvNet, the data association branch and the SOT branch. Data association is treated as a special re-identification task and solved by learning discriminative features for different targets in the data association branch. To handle the problem that the computational cost of SOT grows intolerably as the number of tracked objects increases, we propose an efficient two-stage tracking method in the SOT branch, which utilizes the merits of correlation features and can simultaneously track all the existing targets within one forward propagation. With feature sharing and the interaction between them, data association branch and the SOT branch learn to better complement each other. Using a multi-task objective, the whole network can be trained end-to-end. Compared with state-of-the-art online MOT methods, our method is much faster while maintaining a comparable performance.

Introduction

Online multi-object tracking aims at estimating the locations of multiple objects in the video sequence and yielding their individual trajectories in a sequential manner. It has a wide range of applications in casual video analysis systems such as video surveillance, robot navigation and autonomous driving.

Benefiting from the advances in object detection, the tracking-by-detection paradigm has become popular for MOT in the past decade. Online MOT Methods based on this paradigm mainly focus on associating detection results in each frame provided by a pre-defined object detector with existing tracks, namely the data association problem. However, detection results are not always reliable. Due to the heavy dependency on the performance of the pre-defined object detector, these data association based online MOT methods inherently fail to track some targets in case of missing detection or non-accurate bounding box.

Figure 1: Examples of detection results (left), tracking results using data association (middle) and tracking results of single object tracker (right). The top row shows the case of non-accurate bounding box and the bottom row illustrates the case of missing detection.

To handle this problem, some previous works (Xiang, Alahi, and Savarese 2015; Chu et al. 2017; Zhu et al. 2018) have attempted to introduce single object tracking (SOT) into MOT problem. These works expect to utilize the merits of single object tracker in tracking target by searching for the best matched location without the help of object detections.

However, there are some problems with these methods. First, the data association model and single object tracker used in existing methods (Xiang, Alahi, and Savarese 2015; Zhu et al. 2018) are isolated. It makes the whole MOT algorithm complicated and cumbersome. What's more, separate training data association model and single object tracker can not utilize information from each other. Single object tracker will make redundant efforts in many easy cases (e.g. targets #1 and #5 in Fig. 1) that can be well handled by data association model. Data association model can not obtain better samples from tracking results of single object tracker (e.g. targets #2 and #9 in Fig. 1) for training. In this paper, we integrate data association and SOT into a unified network, which consists of the feature ConvNet, the data association branch and the SOT branch. The two branches can share

*Corresponding author.

convolutional features from the feature ConvNet. Furthermore, we introduce interaction between the two branches to better utilize information from each other during network training. Specifically, we only use targets that are not correctly associated in the data association branch to train the SOT branch, which makes the SOT branch focus on dealing with the cases that the data association fails to handle. Besides, the tracking results obtained from the SOT branch are treated as supplement to detection results, which are also used to train the data association branch.

Another problem is that the tracking speed drops dramatically as the number of tracked objects increases, since existing methods simply apply individual single object tracker for each tracked target. It greatly limits the practical application of these methods. To handle this problem, we propose an efficient two-stage tracking method in the SOT branch, which can simultaneously track all the existing targets within one forward propagation. In the first stage, we introduce local correlation operation on the feature maps of two input frames to compute correlation features for all positions in the feature map and then simultaneously obtain correlation heatmaps for all existing targets using ROI-Pooling. Translation of targets across two input frames can be estimated from these heatmaps. To account for changes in object scale and aspect ratio, the estimated positions of targets are then refined using conventional bounding box regression in the second stage.

To sum up, the contributions of this work are as follows:

First, we set up a unified network for integrating data association and single object tracking, called DASOTNet, which can be trained end-to-end. With the feature sharing and interaction between them, data association and SOT learn to better complement each other.

Second, we propose an efficient two-stage tracking method in the SOT branch of DASOTNet, which can simultaneously track all the existing targets within one forward propagation.

With the proposed DASOTNet, we build an online MOT approach that integrates data association and SOT. Evaluations on challenging MOT16 and MOT17 (Milan et al. 2016) benchmarks demonstrate the effectiveness of the proposed online MOT algorithm.

Related Work

Multi-object Tracking. With the development of object detection methods (Felzenszwalb et al. 2010; Girshick 2015), the tracking-by-detection paradigm has become popular for MOT. Methods (Huang, Wu, and Nevatia 2008; Pirsiavash, Ramanan, and Fowlkes 2011; Milan, Roth, and Schindler 2014; Bae and Yoon 2014) based on this paradigm treat the MOT task as a data association problem and generate trajectories of objects by linking detections across consecutive frames. Data association based MOT methods take advantages of information from multiple detections and targets simultaneously, which can usually obtain better results compared to tracking each target individually. However, the performance of pre-defined object detector is inevitably imperfect. Illumination fluctuation, pose variation and occlusion in crowded scenes may cause unreliable detections, such as false positive, missing detection, and non-accurate bounding box. Most works use detections from both past and future frames to handle noisy detections in batch mode. They consider MOT as a global optimization problem in various forms such as network flow (Pirsiavash, Ramanan, and Fowlkes 2011; Zhang, Li, and Nevatia 2008), continuous energy minimization (Milan, Roth, and Schindler 2014), k-partite graph (Dehghan, Modiri Assari, and Shah 2015), subgraph multi-cut (Tang et al. 2015) and so on. However, methods in batch mode are not suitable for causal applications. On the contrary, data association based online MOT methods (Bae and Yoon 2014; Hong Yoon et al. 2016) link detections to existing targets sequentially, which can only use the information up to the current frame. As a consequence, they can not handle unreliable detections well.

In recent years, single object trackers have been introduced in online MOT task to alleviate the dependency on the quality of detections. For example, Xiang, Alahi, and Savarese apply a simple single object tracker to keep tracking each target individually and resort to data association when the tracking results becomes unreliable. Using a similar pipeline, Zhu et al. adopt a more complicated single object tracker and design a dual matching attention network for data association. Chu et al. design a deep ConvNet to utilize single object tracker throughout the whole tracking process and focus on handling tracking drifts problem with spatial-temporal attention mechanism. However, the data association and single object tracker used in these works are isolated, which can not take full use of the relation between them. Besides, since these methods apply individual single object tracker for each target, they all suffer from the problem that the computational cost grows dramatically as the number of tracked targets increases. In this paper, we propose an online MOT algorithm with a neat network that integrates data association and SOT into a unified framework. With the help of feature sharing and interaction between data association and SOT, they learn to better complement each other. To handle the computation complicity problem when applying SOT to MOT, we propose an efficient two-stage SOT method that can track all the existing targets simultaneously.

Correlation Features. The idea of using local correlation operation on features of two images is originally proposed by Dosovitskiy et al. for estimating optical flow and adopted in other video tasks recently, e.g. video semantic segmentation (Zhu et al. 2017b) and video object detection (Zhu et al. 2017a; Feichtenhofer, Pinz, and Zisserman 2017). The work in D&T (Feichtenhofer, Pinz, and Zisserman 2017) is mostly related to our tracking method in the SOT branch, which also uses correlation features to track objects across frames. However, the manner of utilizing correlation features in our work is quite different from D&T. Correlation features are directly used to regress the target bounding box across frames in D&T, which is a relatively hard problem. While our SOT branch divides this problem into two subproblems and adopts a two-stage method to handle them. First, we use correlation features to obtain the coarse trans-

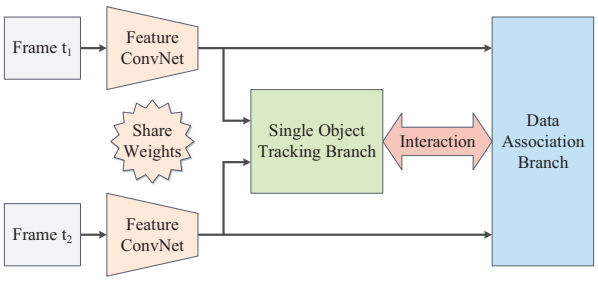


Figure 2: The overall architecture of the proposed DASOTNet. It consists of three modules: feature ConvNet, data association branch and single object tracking branch. These two branches interact with each other during joint training.

lation of the target across frames, which is simpler and more in line with the nature of correlation features. Then we refine the target bounding box using image features by bounding box regression to account for imprecise translation estimation and changes in scale and aspect ratio, which can not be handled by correlation features well. Besides, D&T focus on improving the performance of video object detection aided by tracking, while we aim at solving online multi-object tracking task.

DASOTNet

The overall architecture of the proposed network is illustrated in Fig. 2. Given a set of two input images $\mathbf{I}^{t_1}, \mathbf{I}^{t_2} \in \mathbb{R}^{H\times W\times 3}$ at frame t_1 and t_2, our network first pushes them through a feature ConvNet to compute convolutional feature maps $\mathbf{F}^{t_1}, \mathbf{F}^{t_2} \in \mathbb{R}^{H'\times W'\times C}$ that are shared by tasks of data association and SOT. On top of the shared feature maps, two branches are built for tasks of data association and SOT respectively. We introduce interaction between these two branches during joint training. The whole network can be trained end-to-end.

Feature ConvNet

The feature ConvNet used in our work has a similar architecture to FPN (Lin et al. 2017a). Specifically, we use the truncated ResNet-50 network (He et al. 2016) pre-trained on the ImageNet classification task (Deng et al. 2009) as the backbone network and build the top-down pathway with the outputs of the last three residual blocks. We take the merged feature map of conv3 as the output of the feature ConvNet that has the stride of 8 with respect to the input images.

Data Association Branch

In this work, we consider data association in online MOT as a special re-identification task and solve it by learning discriminative features for representing different targets in the data association branch. Fig. 3 shows the details of the data association branch. Specifically, given the corresponding regions of interest (RoIs) of two input frames, denoted as $\mathcal{R}^{t_1} = \{\boldsymbol{b}_i^{t_1}\}_{i=1}^{N^{t_1}}$ and $\mathcal{R}^{t_2} = \{\boldsymbol{b}_i^{t_2}\}_{i=1}^{N^{t_2}}$ respectively, where $\boldsymbol{b} = (b_x, b_y, b_w, b_h)$ specifies the coordinates of the center of the bounding box and its width and height in pixels, we apply position-sensitive ROI pooling (PSROI-Pooling (Dai et al. 2016)) for each frame to aggregate position-sensitive feature maps, produced from an additional convolutional layer that operates on the output of the shared feature ConvNet. The layer outputs a bank of k^2D position-sensitive feature maps corresponding to a $k \times k$ spatial grid of which each point represents a D-dimensional feature vector at relative positions to be used in the PSROI-Pooling operation. PSROI-Pooled features of each RoI are then global average pooled and L_2-normalized to obtain the final feature representation $\boldsymbol{f}(\boldsymbol{b}) \in \mathbb{R}^D$ for each RoI. After that, we calculate the similarity matrix $\mathbf{S} \in \mathbb{R}^{N^{t_1}\times N^{t_2}}$ among RoIs of two frames as:

$$\mathbf{S} = [s_{ij}]_{N^{t_1}\times N^{t_2}},\ s_{ij} = \langle \boldsymbol{f}(\boldsymbol{b}_i^{t_1}), \boldsymbol{f}(\boldsymbol{b}_j^{t_2})\rangle, \tag{1}$$

where $\langle\cdot,\cdot\rangle$ stands for inner-product of two vectors. Since the feature we used is L_2-normalized, it is equal to the cosine similarity.

We aim at learning discriminative features for representing different targets such that the cosine similarity is high for the same target while low for different targets, which can be treated as a binary classification problem. We scale the cosine similarity s_{ij} with bias to obtain the corresponding classification score s'_{ij} as:

$$s'_{ij} = \begin{cases} a_1(s_{ij} - b), & s_{ij} - b \geq 0 \\ a_2(s_{ij} - b), & s_{ij} - b < 0, \end{cases} \tag{2}$$

where a_1, a_2 and b are pre-defined parameters. We set $a_1 = 10, a_2 = 4, b = 0.7$ in our experiments.

Let $y_{ij} \in \{0, 1\}$ and $p_{ij} = 1/(1+\exp(-s'_{ij})) \in [0, 1]$ be the ground-truth label and the estimated probability for the class with label 1, respectively. We design a variant of focal loss (Lin et al. 2017b):

$$\begin{aligned} L_{DA}(\mathbf{S}) &= -\sum_i^{N^{t_1}}\sum_j^{N^{t_2}} w_{ij}\log(p'_{ij}) \\ w_{ij} &= \frac{\exp(\beta(1-p'_{ij}))}{\sum_m^{N^{t_1}}\sum_n^{N^{t_2}}\exp(\beta(1-p'_{mn}))} \\ p'_{ij} &= \begin{cases} 1-p_{ij}, & y_{ij}=0 \\ p_{ij}, & otherwise, \end{cases} \end{aligned} \tag{3}$$

where $\beta \geq 0$ is a hyper-parameter that adjusts the relative importance of different samples (we set $\beta = 5$ in our experiments).

Single Object Tracking Branch

Directly applying existing single object trackers to MOT task needs to consecutively add new trackers into the system as new targets appear, which causes the tracking speed to slow down intolerably as the number of tracked targets increases. To handle this problem, we propose an efficient tracking method, containing two stages: correlation tracking and position refinement, in the SOT branch, which can simultaneously track all the existing targets within one forward propagation of the network. Fig. 4 shows the details of the SOT branch.

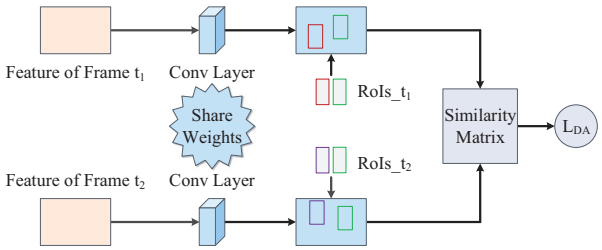


Figure 3: The details of the data association branch.

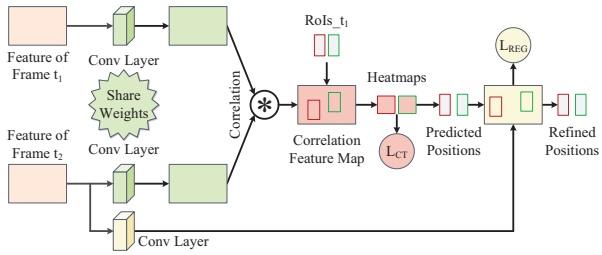


Figure 4: The details of the single object tracking branch.

Correlation Tracking. At the first stage, we first compute correlation feature maps $\mathbf{X}_{corr}^{t_1,t_2}$ [1] for all positions between the feature maps $\mathbf{X}^{t_1}, \mathbf{X}^{t_2} \in \mathbb{R}^{H' \times W' \times C'}$ of two input frames, produced by an extra convolutional layer with the output L_2-normalized along the channel dimension. This layer operates on the output of the shared feature ConvNet. It's cumbersome and unnecessary to take all possible circular shifts in the feature map into consideration, since displacements of the same targets across two adjacent frames is limited. Therefore, we only conduct the correlation operation in a local square neighbourhood with the maximum displacement d as:

$$\mathbf{X}_{corr}(i,j,p,q) = \langle \mathbf{X}^{t_1}(i,j), \mathbf{X}^{t_2}(i+p, j+q) \rangle, \quad (4)$$

where $p, q \in [-d, d]$ are offsets around the location (i, j) in the feature map. Thus the correlation feature map $\mathbf{X}_{corr}$ is of size $H' \times W' \times (2d+1) \times (2d+1)$, which is then reshaped to the size of $H' \times W' \times (2d+1)^2$. Note that the correlation operation is conducted on two L_2-normalized feature maps in our work, the feature vector at the location (i, j) in the correlation feature map $\mathbf{X}_{corr}$ actually encodes the cosine similarity between the feature at the location (i, j) in $\mathbf{X}^{t_1}$ and the feature at each location of a local neighborhood around location (i, j) in $\mathbf{X}^{t_2}$.

We then adopt ROI-Pooling with RoIs $\mathcal{R}^{t_1}$ on the correlation feature map to simultaneously obtain the ROI-Pooled correlation feature map $\mathbf{x}_i \in \mathbb{R}^{k \times k \times (2d+1)^2}$ for each RoI $\boldsymbol{b}_i^{t_1}$ in the frame t_1. The ROI-Pooled correlation feature map $\mathbf{x}_i$ is then spatial global average pooled and reshaped to a heatmap $\mathbf{H}_i \in \mathbb{R}^{(2d+1) \times (2d+1)}$. The displacement of RoI $\boldsymbol{b}_i^{t_1}$ from frame t_1 to frame t_2 can be estimated by the location of the maximum in the heatmap $\mathbf{H}_i$. Learning the heatmap is formulated as a binary classification problem.

[1] Hereafter we ignore the superscript t_1, t_2 unless it is needed.

For each heatmap $\mathbf{H}_i$, there is only one positive location with label 1 in its ground-truth heatmap $\mathbf{H}_i^*$, and all other locations are negative. During training, we treat locations within a local neighborhood of the ground-truth positive location as also positive, but with soft labels. The range of the local neighborhood is determined based on the size of the RoI by ensuring that the bounding box with the displacement within the neighborhood would have at least 0.7 IoU (intersection over union) with the ground-truth bounding box. We then assign soft labels to the locations within the local neighborhood by an unnormalized 2D Gaussian, $\exp(-\frac{1}{2}(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}))$, with the center at the ground-truth location. σ_x and σ_y are set to $1/6$ of the horizontal and vertical range of the local neighborhood respectively. During training, similar problems are also encountered as training the similarity matrix in the data association branch (Sec.). Similarly, we scale the heatmap with bias to obtain the corresponding classification score by applying Eq. 2 to the heatmap and use a variant of focal loss with the augmented ground-truth heatmap:

$$L_{CT}(\mathbf{H}) = -\frac{1}{Z} \sum_{i}^{2d+1} \sum_{j}^{2d+1} w'_{ij} \log(p'_{ij})$$
$$w'_{ij} = \begin{cases} \exp(\beta(1 - p'_{ij})), & y_{ij} = 0 \\ y_{ij}^{\alpha} \exp(\beta(1 - p'_{ij})), & otherwise, \end{cases} \quad (5)$$

where $Z = \sum_i^{2d+1} \sum_j^{2d+1} w'_{ij}$ and α is the hyper-parameter which controls the contribution of each point within the local neighborhood (we set $\alpha = 2$ in our experiments).

Position Refinement. Based on the displacement estimated from the heatmap $\mathbf{H}_i$, we can obtain the predicted position of RoI $\boldsymbol{b}_i^{t_1}$ in frame t_2, denoted as $\hat{\boldsymbol{b}}_i = (\hat{b}_{i,x}, \hat{b}_{i,y}, \hat{b}_{i,w}, \hat{b}_{i,h})$. However, the predicted position $\hat{\boldsymbol{b}}_i$ may be inaccurate in two aspects. First, the estimated displacement is relatively coarse compared to the input image, since the heatmap is obtained from the feature maps with the stride of 8. When we re-map the heatmap to the input image, some precision may be lost. Besides, the predicted position does not account for changes in object scale and aspect ratio.

To handle these problems, we resort to bounding box regression for position refinement. Specifically, an extra convolutional layer is added to the output of the feature ConvNet, which produces a bank of $4k^2$ position-sensitive regression feature maps. Given the predicted position $\hat{\boldsymbol{b}}_i$ of RoI $\boldsymbol{b}_i^{t_1}$, a PSROI-Pooling operation is performed on these regression feature maps to predict the transformation $\boldsymbol{\Delta}_i = (\Delta_{i,x}, \Delta_{i,y}, \Delta_{i,w}, \Delta_{i,h})$ that maps the predicted position $\hat{\boldsymbol{b}}_i$ to the ground-truth box $\boldsymbol{b}_i^*$. $\boldsymbol{b}_i^* = (b_x^*, b_y^*, b_w^*, b_h^*)$ is assigned to the ground-truth box of the RoI at frame t_2 that has the same target ID as RoI $\mathbf{b}_i^{t_1}$.

Following (Girshick 2015), we use the smooth L1 loss to train the transformation as:

$$L_{REG}(\boldsymbol{\Delta}) = \sum_{j \in \{x,y,w,h\}} smooth_{L_1}(\Delta_j^* - \Delta_j)$$
$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & otherwise, \end{cases} \quad (6)$$

where $\mathbf{\Delta}^* = (\Delta_x^*, \Delta_y^*, \Delta_w^*, \Delta_h^*)$ are the ground-truth regression targets defined as:

$$\begin{aligned} \Delta_x^* &= (b_x^* - \hat{b}_x)/\hat{b}_w \quad \Delta_y^* = (b_y^* - \hat{b}_y)/\hat{b}_h \\ \Delta_w^* &= \log(b_w^*/\hat{b}_w) \quad \Delta_h^* = \log(b_h^*/\hat{b}_h), \end{aligned} \tag{7}$$

During online tracking process, the SOT result $\tilde{\boldsymbol{b}}$ is obtained by applying the inverse operation of Eq. 7 to the predicted position $\hat{\boldsymbol{b}}$ with the transformation $\mathbf{\Delta}$.

Joint Training

To make full use of the relation between data association and SOT, we introduce interaction between the two branches during network training. Specifically, we take two frames $\mathbf{I}^{t_1}, \mathbf{I}^{t_2}$ within the frame gap T_0 as inputs. RoIs for frame t_1 are sampled to ensure each of them has at least 0.5 IoU with one of the ground-truth bounding boxes at frame t_1, which are used to simulate the targets at frame t_1. RoIs for frame t_2 are detection results at frame t_2. The target ID of each RoI is assigned by the label of the ground-truth box with the maximum IoU. We first evaluate the data association result between targets at frame t_1 and detections at frame t_2 and push the RoIs of remaining targets that are not correctly associated into the SOT branch. The outputs of the SOT branch, i.e. the SOT results of these remaining targets, are then used as supplement to detections at frame t_2 for training data association branch.

To jointly learn the data association branch and the SOT branch, we use a multi-task loss that consists of L_{DA} in the data association branch, L_{CT} for correlation tracking and L_{REG} for position refinement in the SOT branch. Our network will predict a cosine similarity matrix $\mathbf{S}$, N heatmaps $\{\mathbf{H}_i\}_{i=1}^N$ and N transformations $\{\mathbf{\Delta}_i\}_{i=1}^N$. The overall loss function is defined as:

$$\begin{aligned} L = L_{DA}(\mathbf{S}) &+ \lambda_1 \frac{1}{N_{tra}} \sum_i^N I_i \, L_{CT}(\mathbf{H}_i) \\ &+ \lambda_2 \frac{1}{N_{fg}} \sum_i^N c_i \, L_{REG}(\mathbf{\Delta}_i), \end{aligned} \tag{8}$$

where λ_1, λ_2 are hyper-parameters that control the balance among the three losses (we set $\lambda_1 = \lambda_2 = 1$ in our experiments). I_i is a indicator value for RoI $\mathbf{b}_i^{t_1}$. $I_i = 1$ if there exists target at frame t_2 that has the same target ID as the RoI $\mathbf{b}_i^{t_1}$, otherwise, $I_i = 0$. c_i is also a indicator value for RoI $\mathbf{b}_i^{t_1}$, which indicates that whether the predicted position $\hat{\mathbf{b}}_i$ obtained from the heatmap $\mathbf{H}_i$ of RoI $\mathbf{b}_i^{t_1}$ belongs to foreground. $c_i = 0$ when $I_i = 0$ or the IoU overlap between the predicted position $\hat{\mathbf{b}}_i$ and its corresponding ground-truth box $\mathbf{b}_i^*$ is less than 0.5. Otherwise, $c_i = 1$. $N_{tra} = \sum_i^N I_i$ and $N_{fg} = \sum_i^N c_i$.

Online MOT Algorithm With DASOTNet

Overview

Based on the off-line trained DASOTNet, we propose an online MOT algorithm that integrates the data association and SOT into a unified framework. Given detections in each frame, we first compute the similarity between existing targets and detections and perform data association using Hungarian algorithm (Munkres 1957). For the remaining targets that are not associated due to missing detection or non-accurate bounding box, the corresponding single object tracking results are used as their positions.

Specifically, at each frame t, the inputs to the feature ConvNet are two images $\mathbf{I}^{t-1}, \mathbf{I}^t$ at frame $t-1$ and frame t. The RoIs for these two frames are bounding boxes of all existing targets at frame $t-1$ and all detections at frame t, respectively. Thus, we can obtain the SOT results of all the existing targets at frame $t-1$ and the features of all detections at current frame t with one forward propagation of our DASOTNet. We then compute the affinity matrix $\mathcal{A}$ between targets (including all the existing targets in frame $t-1$ and untracked targets in history frames) and detections in terms of appearance and motion:

$$\begin{aligned} &\mathcal{A} = [a_{ij}], \; a_{ij} = a_{ij}^{app} a_{ij}^{mot}, \; a_{ij}^{app} = < \bar{\mathcal{F}}(\mathcal{O}_i), \boldsymbol{f}(\mathbf{b}_j^{\mathcal{D}}) > \\ &a_{ij}^{mot} = \exp\left(-\frac{1}{2}\left(\left(\frac{\bar{b}_{i,x} - b_{j,x}^{\mathcal{D}}}{b_{j,w}^{\mathcal{D}}}\right)^2 + \left(\frac{\bar{b}_{i,y} - b_{j,y}^{\mathcal{D}}}{b_{j,h}^{\mathcal{D}}}\right)^2\right)\right), \end{aligned} \tag{9}$$

where a_{ij}^{app} and a_{ij}^{mot} indicate appearance and motion affinity between target $\mathcal{O}_i$ and detection $\mathcal{D}_j$, respectively. $\bar{\mathcal{F}}(\mathcal{O})$ represents the feature of target, which is the average of historical features of the target. $\bar{\mathbf{b}} = (\bar{b}_x, \bar{b}_y, \bar{b}_w, \bar{b}_h)$ and $\mathbf{b}^{\mathcal{D}} = (b_x^{\mathcal{D}}, b_y^{\mathcal{D}}, b_w^{\mathcal{D}}, b_h^{\mathcal{D}})$ denote the position of the target predicted by a simple motion model using Kalman filter and the bounding box of detection, respectively. The Hungarian algorithm is applied to the affinity matrix $\mathcal{A}$ with a threshold τ_a for the minimum affinity. After that, we evaluate the similarity of each un-associated target at frame $t-1$ and its corresponding SOT result $\tilde{\boldsymbol{b}}$. The SOT result $\tilde{\boldsymbol{b}}$ will be used as the position of the target if the similarity is higher than τ_s. Otherwise, the target is considered as untracked at frame t.

Target Management

For target initialization, we set a threshold τ_d and only detections with the detection score over τ_d are used. For target termination, we stop tracking the target if it is not associated with any detection over τ_t successive frames or exits the field of view. Besides, to alleviate the influence of false positive detections, we also terminate the target if it is not associated with any detection in any of the first τ_i frames since the target is initialized.

Experiments

Datasets. We evaluate our online MOT algorithm on MOT16 and MOT17 benchmark datasets. MOT16 dataset collects 14 (7 for training, 7 for test) video sequences in unconstrained environments and provides public object detections (DPM (Felzenszwalb et al. 2010)). MOT17 contains the same sequences as MOT16 with more accurate ground truth, in which each sequence is provided with 3

Table 1: The performance on validation set. DA and SOT respectively stand for only using data association and single object tracking. DA+SOT means a combination of separately trained data association and single object tracking during online tracking. DASOT is the proposed method.

Method	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	FPS↑
DA	29.6%	37.4%	27	120	427	**11.8**
SOT	24.4%	25.7%	18	133	1539	10.1
DA+SOT	31.5%	39.4%	32	113	402	6.7
DASOT	**35.7%**	**42.3%**	**34**	**110**	**251**	9.4

sets of detections (DPM (Felzenszwalb et al. 2010), Faster-RCNN (Ren et al. 2015) and SDP (Yang, Choi, and Lin 2016)) for more comprehensive evaluation. The ground truth annotations of training sequences are released. We split training sequences in MOT16 dataset into training set and validation set. The two sets have roughly the same number of frames. We then conduct alation studies on validation set with the DASOTNet trained using split training set. The ground truth annotations of test sequences in these datasets are not released and the tracking results are automatically evaluated by the benchmark. So we use the test sequences with the DASOTNet trained using all the training sequences for comparison with various state-of-the-art MOT methods.

Evaluation Metrics. We use the metrics suggested in MOT16 (Milan et al. 2016) benchmark for evaluation, which includes Multiple Object Tracking Accuracy (MOTA) (Bernardin and Stiefelhagen 2008) that combines False Positives (FP), False Negatives (FN) and the Identity Switches (IDS), ID F1 score (Ristani et al. 2016) (IDF1, the ratio of correctly identified detections over the average number of ground truth and computed detections), the ratio of Mostly Tracked targets (MT) and the ratio of Mostly Lost targets (ML).

Implementation Details. The proposed algorithm is implemented using python with Caffe (Jia et al. 2014). All extra convolutional layers added to the output of feature ConvNet have the same kernel size of 1×1 and the parameters in these layers are Gaussian randomly initialized with $std = 0.01$. k for PSROI-Pooling is set to 7. The dimension of feature for data association D and the channels of feature map used in correlation tracking C' are set to 40. The thresholds τ_a and τ_s are set to 0.6 and 0.7 respectively. The thresholds τ_t and τ_i in target management are set to 30 and 3 respectively. The detection score threshold τ_d for target initialization is set to 0.25, 0.5 and 0.4 for detections from (Felzenszwalb et al. 2010), (Ren et al. 2015) and (Yang, Choi, and Lin 2016), respectively. The frame gap T_0 is set to 10. The maximum displacement d for correlation tracking is set to 25 for training and 10 for testing. For off-line training DASOTNet, we use the SGD optimizer with momentum 0.9, weight decay 5×10^{-4} and learning rate initialized at 10^{-3} and dropped to 10^{-4} after 20k iterations for split training set and 40k iterations for all training sequences.

Performance Analysis

The Impact of Integration. As shown in Table 1, the combination of data association and SOT can improve tracking performance compared to using one of them separately, which demonstrates that data association and SOT can complement each other. Besides, with the help of feature sharing and interaction between data association and SOT joint training, our method achieves better performance and less computation cost (save 45% memory and speed up 40%) compared to combination of separately trained data association and SOT during online tracking, which demonstrates the effectiveness of the proposed DASOTNet.

Table 2: Tracking speed (FPS) and the average number of tracked targets per frame in some sequences of MOT16 test set. Experiments are conducted on a 2.6 GHz CPU and a TITAN Xp.

Seq	MOT16-12	MOT16-14	MOT16-07	MOT16-03
FPS	9.0	8.9	8.5	7.6
Num	9.2	24.6	32.6	69.7

The Impact of Two-Stage SOT Method. We also conduct several experiments to demonstrate the contribution of the proposed two-stage tracking method in the SOT branch. First, we disable the SOT module and only use the data association module to track targets, which is the baseline algorithm. We set the state of the target as lost and leave the position of the target at frame t as empty if it is not associated with any detections at frame t in the baseline algorithm. Denote l as the maximum number of frames allowed for SOT module adding the SOT results. In other words, if a target is not associated in data association from frame t to frame $t + L$ ($L \geq l$), the SOT results of the target from frame $t+l$ to frame $t+L$ can not be added as the position of the target. We then gradually add SOT results with different tracking methods via increasing l to compare their impacts on the performance in terms of MOTA. The experimental results are shown in Fig. 5. We can see that our two-stage tracking method performs best compared to D&T (Feichtenhofer, Pinz, and Zisserman 2017) and a simple method based on Kalman Filter. Note that the performance of MOTA is persistently improved as l increases using our method, which demonstrates that our method can better utilize correlation features than D&T (Feichtenhofer, Pinz, and Zisserman 2017) and is more suitable for MOT.

Table 2 shows the tracking speed of the proposed method and the average number of tracked targets per frame in some sequences of MOT16 test set. As shown in the table, the number of tracked targets has little effect on the tracking speed of our method, which demonstrates the efficiency of the proposed two-stage SOT method.

Evaluation on MOT Benchmarks

We evaluate our algorithm, denoted by DASOT, on the test sequences of MOT16 and MOT17 benchmarks against several state-of-the-art online MOT methods. All the compared methods and ours use the same public detections provided by the benchmark. For fair comparison, we do NOT use the bounding box regression to modify the original public detection, although it can improve performance. In all the experiments, the bounding box regression is only used in the

Table 3: Quantitative results of our method (denoted by DASOT) and several state-of-the-art online MOT trackers on MOT16 and MOT17 test sequences. Values in bold highlight the best results. The arrows indicate low or high optimal metric values.

Dataset	Method	MOTA↑	MT↑	ML↓	FP↓	FN↓	IDS↓	IDF1↑	FPS↑
MOT16	CDA_DDAL (Bae and Yoon 2017)	43.9%	10.7%	44.4%	6450	95175	676	45.1%	0.5
	DCCRF (Zhou et al. 2018)	44.8%	14.1%	42.3%	5613	94133	968	39.7%	0.1
	RAR (Fang et al. 2018)	45.9%	13.2%	41.9%	6871	91173	648	48.8%	0.9
	STAM(Chu et al. 2017)	46.0%	14.6%	43.6%	6895	91117	**473**	50.0%	0.2
	DMAN (Zhu et al. 2018)	46.1%	**17.4%**	42.7%	7909	89874	532	**54.8%**	0.3
	AMIR (Sadeghian, Alahi, and Savarese 2017)	**47.2%**	14.0%	**41.6%**	**2681**	92856	774	46.3%	1.0
	DASOT (ours)	46.1%	14.6%	**41.6%**	8222	**89204**	802	49.4%	**9.0**
MOT17	GMPHD_KCF (Kutschbach et al. 2017)	39.6%	8.8%	43.3%	50903	284228	5811	36.6%	3.3
	SAS (Maksai and Fua 2019)	44.2%	16.1%	44.3%	29473	283611	**1529**	**57.2%**	4.8
	DMAN (Zhu et al. 2018)	48.2%	19.3%	38.3%	26218	263608	2194	55.7%	0.3
	HAM_SADF (Yoon et al. 2018)	**48.3%**	17.1%	41.7%	**20967**	269038	1871	51.1%	5.0
	DASOT (ours)	48.0%	**19.9%**	**34.9%**	38830	**250533**	3909	51.3%	**9.1**

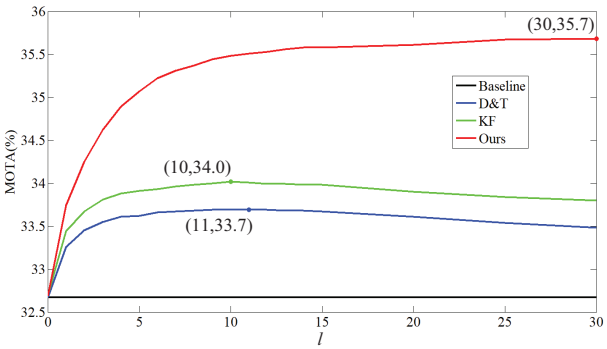
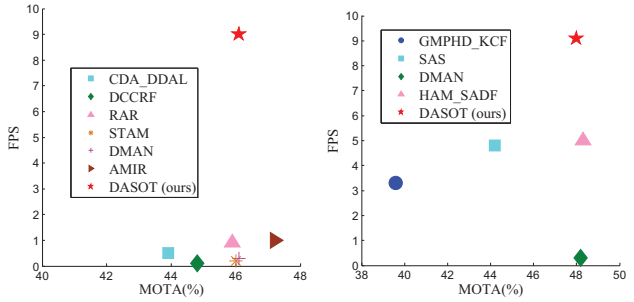


Figure 5: The performance curve of different single object tracking methods in terms of MOTA.



(a) MOTA VS FPS on MOT16. (b) MOTA VS FPS on MOT17.

Figure 6: Tracking performance (MOTA) and tracking speed (FPS) of the proposed method and other state-of-the-art online MOT methods on MOT16 and MOT17 datasets. Each marker indicates a tracker. Higher and more right is better.

single object tracking branch.

Table 3 presents the quantitative comparison results. Overall, our algorithm DASOT achieves a comparable MOTA score with the state-of-the-art online methods on both benchmarks with a much faster tracking speed. Since our algorithm uses single object tracking result as the position of the target that is not associated by data association, it will introduce more FP error. On the contrary, the number of FN, MT and ML will be reduced. As expected, DASOT performs best in terms of ML and FN on both benchmarks and achieves the best performance in MT metric on MOT17 benchmark. In terms of IDS and IDF1, our method performs worse than state-of-the-art online methods (Chu et al. 2017; Zhu et al. 2018), mainly due to that we do not specially design the attention mechanism for handling occlusion as these methods.

To better illustrate the effectiveness of the proposed algorithm, we visualize the relation between tracking performance (MOTA) and tracking speed (FPS) of our algorithm and several state-of-the-art online MOT methods. As shown in Fig. 6, our algorithm strikes a better balance between accuracy and speed compared to other online MOT methods.

Conclusion

In this work, we integrate data association and single object tracking into a unified network DASOTNet, which can be trained end-to-end. With the help of feature sharing and interaction between data association and single object tracking, they learn to better complement each other. To handle the tracking speed problem when applying single object tracking to MOT, we design an efficient two-stage tracking method, which utilizes the merits of correlation features and can simultaneously track all the existing targets in one forward propagation. With the offline trained DASOTNet, we build an online MOT algorithm and demonstrate its effectiveness on public MOT benchmarks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61371192), the Key Laboratory Foundation of the Chinese Academy of Sciences (CXJJ-17S044), the Fundamental Research Funds for the Central Universities (WK2100330002, WK3480000005) and Major Scientifc Research Project of Zhejiang Lab (No. 2019DB0ZX01).

References

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

Bae, S.-H., and Yoon, K.-J. 2017. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *TPAMI*.

Bernardin, K., and Stiefelhagen, R. 2008. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*.

Chu, Q.; Ouyang, W.; Li, H.; Wang, X.; Liu, B.; and Yu, N. 2017. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *ICCV*.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*.

Dehghan, A.; Modiri Assari, S.; and Shah, M. 2015. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *ICCV*.

Fang, K.; Xiang, Y.; Li, X.; and Savarese, S. 2018. Recurrent autoregressive networks for online multi-object tracking. In *WACV*.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2017. Detect to track and track to detect. In *ICCV*.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *TPAMI*.

Girshick, R. 2015. Fast r-cnn. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hong Yoon, J.; Lee, C.-R.; Yang, M.-H.; and Yoon, K.-J. 2016. Online multi-object tracking via structural constraint event aggregation. In *CVPR*.

Huang, C.; Wu, B.; and Nevatia, R. 2008. Robust object tracking by hierarchical association of detection responses. In *ECCV*.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Kutschbach, T.; Bochinski, E.; Eiselein, V.; and Sikora, T. 2017. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In *AVSS*.

Lin, T.-Y.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017a. Feature pyramid networks for object detection. In *CVPR*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollar, P. 2017b. Focal loss for dense object detection. In *ICCV*.

Maksai, A., and Fua, P. 2019. Eliminating exposure bias and metric mismatch in multiple object tracking. In *CVPR*.

Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.

Milan, A.; Roth, S.; and Schindler, K. 2014. Continuous energy minimization for multitarget tracking. *TPAMI*.

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*.

Pirsiavash, H.; Ramanan, D.; and Fowlkes, C. C. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*.

Sadeghian, A.; Alahi, A.; and Savarese, S. 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *ICCV*.

Tang, S.; Andres, B.; Andriluka, M.; and Schiele, B. 2015. Subgraph decomposition for multi-target tracking. In *CVPR*.

Xiang, Y.; Alahi, A.; and Savarese, S. 2015. Learning to track: Online multi-object tracking by decision making. In *ICCV*.

Yang, F.; Choi, W.; and Lin, Y. 2016. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*.

Yoon, Y.-c.; Boragule, A.; Song, Y.-m.; Yoon, K.; and Jeon, M. 2018. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. In *AVSS*.

Zhang, L.; Li, Y.; and Nevatia, R. 2008. Global data association for multi-object tracking using network flows. In *CVPR*.

Zhou, H.; Ouyang, W.; Cheng, J.; Wang, X.; and Li, H. 2018. Deep continuous conditional random fields with asymmetric inter-object constraints for online multi-object tracking. *TCSVT*.

Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017a. Flow-guided feature aggregation for video object detection. In *ICCV*.

Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017b. Deep feature flow for video recognition. In *CVPR*.

Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; and Yang, M.-H. 2018. Online multi-object tracking with dual matching attention networks. In *ECCV*.