Multiple Object Tracking in Videos Based on LSTM and Deep Reinforcement Learning

Ming-xin Jiang [1,2], Chao Deng [3]*, Zhi-geng Pan [4,5]*, Lan-fang Wang[2], Xing Sun[2]

1 Jiangsu Laboratory of Lake Environment Remote Sensing Technologies, Huaiyin Institute of Technology, Huaian, China , 223003；
2 Faculty of Electronic information Engineering, Huaiyin Institute of Technology, Huaian, China, 223003；
3 School of Physics & Electronic Information Engineering, Henan Polytechnic University,Jiaozuo,China, 454000 ;
4 Institute of Industrial VR, Foshan University, Foshan, China，528000；
5 Digital Media &Interaction Research Center, Hangzhou Normal University, Hangzhou, China, 310012；

* Correspondence: super@hpu.edu.cn, and zgpan@hznu.edu.cn

Abstract: Multiple object tracking is a challenging issue in computer vision community. In this paper, we propose a multi-object tracking algorithm in videos based on Long Short-Term Memory (LSTM) and deep reinforcement learning. Firstly, the multiple objects are detected by the object detector YOLO V2. Secondly, the problem of single object tracking is considered as a Markov decision process (MDP) since this setting provides a formal strategy to model an agent that makes a sequence decisions. The single object tracker is composed of a network that includes a CNN followed by an LSTM unit. Each tracker, regarded as an agent, is trained by utilizing deep reinforcement learning. Finally, we conduct a data association using LSTM for each frame between the results of objects detector and the results of single object trackers. From the experimental results, we can see that our tracker achieves better performance than the other state-of-art methods. Multiple targets can be steadily tracked when frequent occlusion, similar appearance, scale changes happened.

Keywords: Multi-object tracking; data association; LSTM; Deep Reinforcement Learning

1. Introduction

Multi-object tracking in videos plays an important role in a wide range of applications, for example, video surveillance, robot navigation, intelligent transportation systems, video analysis, to name a few [1-2]. Despite the field has made tremendous progress since the early work, visual multi-object tracking is still regarded as a challenging problem due to frequent occlusions, appearance similarity between objects, varying number of objects, environmental noise within measurements [3-4].

1.1 Related Works

Tracking-by-detection methods[5-7] have appeared as one of the most successful strategies due to recent advances in methods for object detection[8-10]. Most of recent tracking–by-detection algorithms that aim to decompose multi-object tracking into two stages: object detection and data association. These algorithms apply the object detector in each frame and associate the results of the detector continuously. Therefore, this kind of multi-object tracking method can recognize emerging or disappearing objects in the sequences of a video more easily, and the search space of objects hypothesis can be greatly reduced.

Tracking-by-detection methods are frequently classified roughly into two categories: offline approaches and online approaches. Offline approaches often use the detections of all the frames of the video sequence together to build long trajectories against false detections and occlusion. A crowded or cluttered scene usually causes some detection failures, which will decrease the accuracy of data association in turn. To compensate for these problems, many multi-object tracking

algorithms using the global data association have been proposed [11-14].However, the performance of the offline approaches is still limited, and it is hard to apply the offline approaches to real-time applications. As data associations between detections and trackers for each frame are performed in an online manner, we can apply the online methods to real-time applications. Seung-Hwan et al. [15] proposed a novel online visual multi-object tracking approach that can handle the similarity between multiple objects.

Data association is the major issue of tracking-by-detection methods[16]. Classical data association approaches include the Joint Probabilistic Data Association Filter (JPDAF) and Multi-Hypotheses Tracking (MHT)[17]. JPDAFs consider all possible associations between objects to make the best assignment in each time step. MHT considers multiple possible associations over several time steps, but its application can be usually limited due to its complexity. Many recent multi-object tracking algorithms have concentrated on enhancing the performance of the object detector or designing a better data association schemes [18-20].

In recent years, LSTM has attracted increasing attention in modeling sequential data. The applications cover feature selection [21], machine translation [22], action recognition [23], video captioning[24], human trajectory prediction [25], etc. The main advantages of LSTMs for modeling sequential data is that they allow end-to-end fine-turning and they are not confined to fixed length inputs to outputs. Inspired by the successful works that have applied LSTM in computer vision fields, we adopt a data association method based on LSTM in this paper. LSTM includes non-linear transformations and memory cells, which makes it effective for the data association.

Most previous multi-object tracking methods represent objects using raw pixel and low-level hand-crafted feature, such as histograms of oriented gradients (HOG)[26], Harr-like features [27], and local binary patterns (LBP)[28]. Although they achieve computational efficiency, they have many limits because hand-crafted features can not capture more complex characteristics of the objects. Recently, deep learning has received much attentions with state-of-the-art results in complicated tasks such as object detection [29], image classification [30], object recognition [31], object tracking [32]. A deep learning tracker (DLT) was proposed in [33]，which uses stacked de-noising auto-encoder to learn the generic features from a large number of auxiliary images offline. However, DLT tracker cannot describe the temporal invariance of deep features, which is important for visual object tracking. [34] developed a deep learning tracking method that uses a two-layer convolutional neural network (CNN) to learn hierarchical features from auxiliary video sequences, in the visual tracking method, appearance variations and complicated motion transformations of objects are taken into account. In [35], the authors present a visual tracking algorithm, which includes a specific feature extractor with CNNs from an offline training set, both spatial and temporal features can be learned by the CNNs jointly from image pairs of two adjacent frames. These deep learning trackers often overlook how to search the interesting region of objects and select the best candidate as the tracking result.

With recent exciting achievements of deep learning, integrating deep learning methods with RL has recently shown very promising results on decision-making problems, i.e. Deep Reinforcement Learning (DRL). Deep neural networks are able to make reinforcement learning algorithms perform more effective because they can provide deep feature representations. DRL algorithms have achieved unequalled success in many challenging domains, e.g., Atari games [36], playing board game GO [37]. In the computer vision community, there are also many attempts of applying DRL to solve traditional tasks, such as action recognition [38], object localization [39], object tracking [40] and region proposal [41]. Luo et al. propose an end-to-end active object tracking algorithm via reinforcement learning, which addresses tracking and camera control simultaneously [42]. In [43], the authors present an action-decision networks for visual tracking with deep reinforcement learning. However, these tracking methods based on deep reinforcement learning usually focus on single object, there is little work related to multi-object tracking. Unlike the aforementioned methods, our method exploits how to apply deep reinforcement learning to solve the online multi-object tracking problem.

1.2 Summary of Contributions

Our motivation is to design a real-time multiple object tracker via LSTM and DRL, which can incorporating appearance by DRL and learning a more effective association strategy by LSTM to improve the performance of tracking. The key contributions of this paper can be summarized as follows:

- We propose a novel visual multi-object tracking algorithm based on LSTM and deep reinforcement learning to solve the problems in the existing methods, which is model-free and requires no prior knowledge. To the best of our knowledge, we are the first to combine such concepts to overcome problems in the process of the visual multi-object tracking.
- The proposed multi-object tracker includes three modules: an object detection module, a number of single object trackers, a data association module. We adopt YOLO V2 as object detector as it is a real-time detection system. Each single object tracker is treated as an agent, which is trained using DRL. A LSTM-based architecture is adopted to solve joint data association problem.
- To compare our multi-object tracker with other state-of-the-art methods qualitatively and quantitatively, we conducted extensive experiments on publicly available challenge benchmark datasets.

The rest of our paper is structured as follows: the following section reviews the background. Section 3 introduces the proposed multi-object tracking framework. Section 4 demonstrates the experimental results and analysis. Finally, we draw conclusions in Section 5.

2. Background

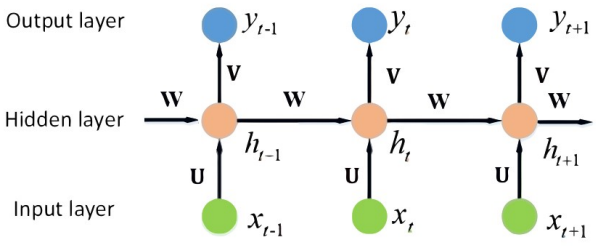
2.1 Long Short-Term Memory(LSTM)

Traditional recurrent neural networks (RNNs) contain cyclic connections that make them a powerful tool to learn complex temporal dynamics, as shown in Figure 1. The formulas that govern the computation happening in a RNN are as follows:

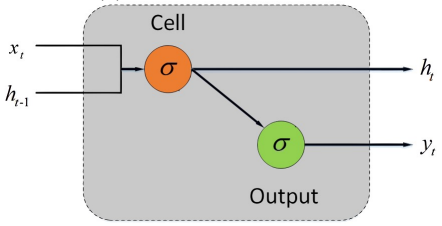
$$h_t = f(\mathbf{U}x_t + \mathbf{W}h_{t-1}) \tag{1}$$

$$y_t = \text{soft}\max(\mathbf{V}h_t) \tag{2}$$

where f is an element-wise non-linearity function, x_t and y_t represent the input vector and the output vector at time step t, $h_t \in \mathbb{R}^N$ is the hidden layer vector with N hidden units at time step t. $\mathbf{U}, \mathbf{W}, \mathbf{V}$ are the weight matrices of the connection from input nodes to hidden nodes, hidden nodes to hidden nodes, and hidden nodes to output nodes.



(a) RNN architecture



(b) RNN Unit

Figure 1. Recurrent neural networks

Though RNNs have been successfully used for sequence modeling tasks, they can only model the data within a fixed-size window. At the same time, training conventional RNNs is difficult due to the exploding and vanishing gradients problem. These problems limit the capability of RNNs to learn long-term dynamics. LSTM was proposed in [44] to solve these problems. The LSTM unit is used in this paper as described in [45], as shown in Figure 2.

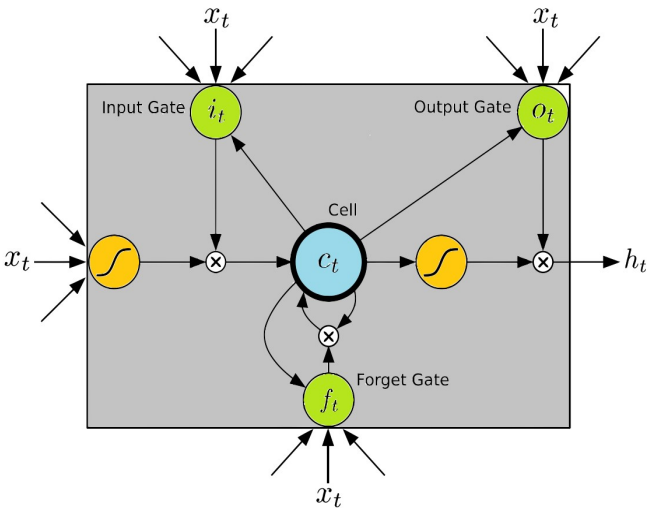


Figure 2. Long Short-term Memory unit

In this subsection, we provide the equations of LSTM for a single memory unit only. Let $\mathbf{x} = (x_1, ..., x_T)$ an input sequence, $\mathbf{y} = (y_1, ..., y_T)$ represents an output sequence, a LSTM network computes a mapping iteratively between $\mathbf{x} = (x_1, ..., x_T)$ and $\mathbf{y} = (y_1, ..., y_T)$ using the following equations:

$$i_t = \sigma(\mathrm{W}_{xi}\, x_t + \mathrm{W}_{hi}\, h_{t-1} + \mathrm{W}_{ci}\, c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(\mathrm{W}_{xf}\, x_t + \mathrm{W}_{hf}\, h_{t-1} + \mathrm{W}_{cf}\, c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(\mathrm{W}_{xc}\, x_t + \mathrm{W}_{hc}\, h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(\mathrm{W}_{xo}\, x_t + \mathrm{W}_{ho}\, h_{t-1} + \mathrm{W}_{co}\, c_t + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(c_t) \quad (7)$$

where $\sigma(x) = 1/(1 + exp(-\mathrm{x}))$ is the logistic sigmoid function, c_t is the cell input activation vectors, i_t describes the input gate, f_t represents the forget gate, o_t output gate. All of above are the same size as the hidden vector h_t .That is, in addition to a hidden vector $h_t \in \mathbb{R}^N$, the LSTM includes an input gate $i_t \in \mathbb{R}^N$, forget gate $f_t \in \mathbb{R}^N$, output gate $o_t \in \mathbb{R}^N$ and memory cell $c_t \in \mathbb{R}^N$. We can find the meaning of the weight matrix, such as W_{hi} represents the hidden to input gate matrix, W_{xo} represents the input to output gate matrix, etc. b_i, b_f, b_o, b_c are the bias terms which are added to i, f, o and c.

2.2 Deep Reinforcement Learning (DRL)

Reinforcement Learning (RL) can be used to solve sequential decision making problems usually. The process of a Reinforcement Learning is as shown in Figure 3. Recently, significant progress has been made by combining reinforcement learning with the ability for learning feature representations in deep learning. Deep Q Network (DQN) and policy gradient are two popular

methods in DRL algorithms. DQN is a form of Q-learning with function approximation using a neural network, which means it tries to learn a state-action value function Q given by a neural network in DQN by minimizing temporal-difference errors. To improve performance and keep stability, various network architectures based on the DQN algorithm such as Dueling DQN [46] and Double DQN [47].

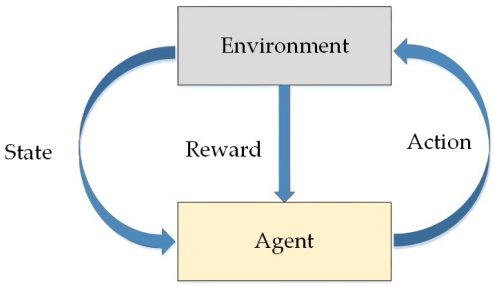


Figure 3. The typical framing of a reinforcement learning

A policy gradient approach is a type of reinforcement learning method that directly optimizes parametrized policies by using gradient descent [48]. Policy gradient methods have many advantages compared to traditional reinforcement learning approaches. For example, they need fewer parameters to represent the optimal policy than the corresponding value function, they do not suffer from the difficult problem causing by uncertain state information, etc.

3. Proposed Visual Multi-object Tracking Algorithm

In the following subsections, we show a brief architecture of our proposed multi-object tracking algorithm firstly. The details of our method are described in the following content.

3.1 Architecture of the proposed multi-object tracking algorithm

Our method consists of three major components: an object detection module, many single object trackers and a data association module, which are shown in Figure 4. In the first place, as demonstrated in Figure 4, we choose YOLO V2 [49] as object detector because it is a state-of-the-art, real-time object detection system. YOLO V2 is applied on every frame and outputs a set of detections D_t at time step t. In each frame, YOLO V2 may output many kinds of detections. To obtain the correct detections to the tracking objects, the intersecton-over-union (IoU) distance is computed between the ground truth and the detections at first frame. And the IoU distance between the mean of its short-term history of validated detections and the current detections is computed to obtain the correct detections at other frame. In the second place, the single object tracker is composed of a network that includes a CNN followed by an LSTM unit. Each tracker, regarded as an agent, is trained by utilizing deep reinforcement learning. Finally, inspired by [50], we adopt a LSTM-based architecture that can learn to solve joint data association problem from training data.

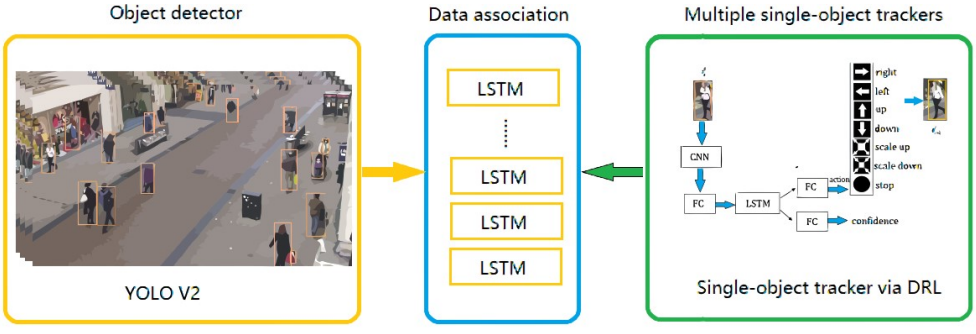


Figure 4. Overview of our proposed multi-object tracking algorithm

3.2 Single-object tracker via deep reinforcement learning

We cast the problem of object tracking as a Markov decision process (MDP) since this setting provides a formal strategy to model an agent that makes a sequence decisions. In our formulation, a single frame image is considered as the environment, in which the agent transforms a bounding box using a set of actions. The MDP includes a set of actions $a \in \mathcal{A}$, a set of states $s \in \mathcal{S}$, a state transition function $f(s,a)$, and a reward signal r. Our single-object tracking framework is illustrated as Figure 5, this section presents details of these components.

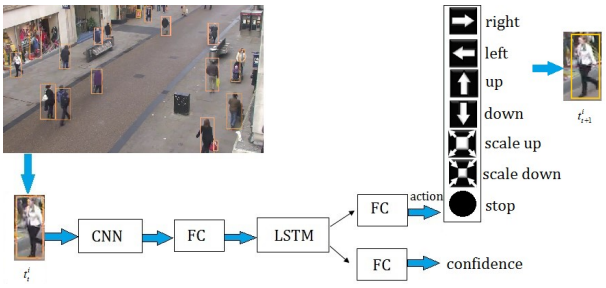


Figure 5. The pipeline of the single-object tracker

In our paper, the set of action $\mathcal{A}$ is composed of six actions that can be applied to the bounding box and one action to terminate the search process, as shown in figure 6, each action is encoded by the 7-dimensional vector. These actions are organized in three sub-sets: horizontal moves {right, left}, vertical moves {up, down}, scale changes {scale up, scale down}.

The state definition is a tuple $s_t=(p_t, v_t)$, where p_t is the image patch (which is pointed by a 4-dimensional vector $p_t=[x_t, y_t, w_t, h_t]$) within the bounding box of the object, v_t is a vector with the history of taken actions. The history vector stores past 10 actions, which means v_t has 70 dimension as each action vector has 7 dimension. At time step t+1, the state $s_{t+1}=(p_{t+1}, v_{t+1})$ is decided by $s_t=(p_t, v_t)$ and the state transition functions, where $p_{t+1} = f_t(p_t, a_t)$, $v_{t+1}=f_v(v_t, a_t)$.

The agent will receive a reward signal r_t from the environment during the training process. In our method, reward r_t is given at the end of a tracking episode when the object is tracked successfully. More specifically, the reward signal $r_t=0$ during iteration in MDP in a time step. When 'stop' action is selected at termination step T, the reward signal r_T is a thresholding function of IoU as follows:

$$r_T = \begin{cases} 1 & if \quad IoU(p_T, \mathrm{g}) > \tau \\ -1 & otherwise \end{cases} \tag{8}$$

where $IoU(p_T, \mathrm{g}) = \mathrm{area}(p_T \cap g) / \mathrm{area}(p_T \cup g)$ represents overlap ratio of p_T and the ground truth of the object.

We adopt policy-based reinforcement learning methods as it has better capability of learning random policies and convergence properties. Our whole network is parameterized by W, the policy-based method models the policy function $\pi(a|\mathrm{s};\mathrm{W})$ and the value function $\mathrm{V}(a|\mathrm{s};\mathrm{W}')$, the aim of training this network is to maximize the overall tracking performance by policy gradient approximation. At each time step t, the goal of the agent is to learn a policy function $\pi(a_t|\mathrm{s}_t;\mathrm{W})$. Approximation of the policy function can be obtained by stochastic gradient ascent algorithm. As there are very limited amount of labelled data for multi-object tracking, we use synthetic data as supplementary to the real data in the training. The parameters W and W' can be learn according to the following equations:

$$\mathrm{W} \leftarrow \mathrm{W}+\lambda(R_t-\mathrm{V}(a_t|\mathrm{s}_t;\mathrm{W}'))\nabla_W \log\pi(a_t|\mathrm{s}_t;\mathrm{W})+\varepsilon\nabla_W H(\pi(\cdot|\mathrm{s}_t;\mathrm{W})) \quad (9)$$

$$\mathrm{W}' \leftarrow \mathrm{W}'-\lambda(R_t-\mathrm{V}(a_t|\mathrm{s}_t;\mathrm{W}') \quad (10)$$

where $R_t=\sum_{t'=t}^{t+T-1}\alpha^{t'-t}r_{t'}$ is the sum of future rewards up to T time steps, $0<\alpha\le 1,\lambda$ is the learning rate, $H(\cdot)$ is an entropy regularizer, ε is the regularizer factor.

Our deep CNN is conducted on the VGG-16 network, which includes five pooling stages, i.e. Conv1-2, Conv2-2, Conv3-3, Conv4-3, Conv5-3. The gradual decrease in the spatial resolution occurs when the depth of layers increases, because all convolutional layers have a 2×2 kernel size and a stride of 2 in the VGG-16 model. For example, when inputting an image with size $M\times N$, the output feature maps of pooling 5 have size $\frac{M}{2^5}\times\frac{N}{2^5}$. In our model, we use the feature maps from Conv3-3, Conv4-3 and Conv5-3, which have been elevated to the same size by using bilinear interpolation.

3.3 Data association

Let $P_t=\{p_t^i\}_{i=1}^M$ represent the set of all outputs of single-object trackers at time step t, p_t^i refers to the state of the i-th output of single-object tracker and M is the number of objects that can be tracked simultaneously in one time step. The state of the i-th object is represented by 4-dimensional vector $p_t^i=[x_t^i,y_t^i,w_t^i,h_t^i]$.We define $\mathrm{Q}_t=\{q_t^j\}_{j=1}^N$ as the set of detections from the object detector with q_t^j the j-th detection and N the number of detections. Let $D_t\in\mathbb{R}^{M\times N}$ denote the similarity matrix for data association measures the relation between an output of single-object tracker p_t^i and a detection m_t^j, where $D_t^{ij}=\left\|p_t^i-q_t^j\right\|_2$ is the Euclidean distance between the p_t^i and q_t^j. Data association based on LSTM for object i is illustrated in Figure 6.

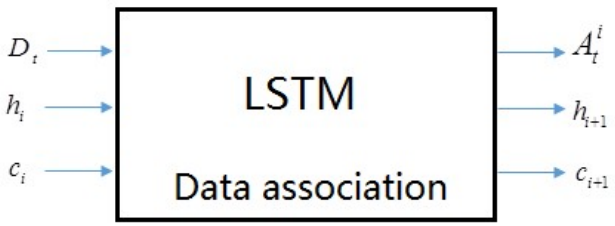


Figure 6. LSTM-based architecture for data association

The task of data association is to predict the assignment for each object using the temporal step-by-step functionality of LSTM. The inputs at each step i are the hidden state h_i, the cell state c_i,and the similarity matrix D_t.The output are the hidden state h_{i+1}, the cell state c_{i+1},and the assignment probability vector A_t^i. A_t^i is a vector of assignment probabilities for object i and all available measurements, which is obtained by applying a softmax layer with normalization to the predicted values. $A_t^{ij}=a$ (object i assigned to the j-th detection) and $\sum_j A_t^{ij}=1$. Let ε is the correct assignment, we adapt negative log-likelihood loss as the cost function to measure the misassignment cost:

$$C(A_t^i,\varepsilon)=-\log(A_t^{i\varepsilon}) \quad (11)$$

The data association requires more representation power, so it is a more complex task. The data association module based LSTM include two layers and 512 hidden units. It takes approximately 40 hours to train all the modules in our tracker on a CPU. The training can be sped up significantly by using of GPUs.

4. Experiments

4.1 Qualitative Evaluation

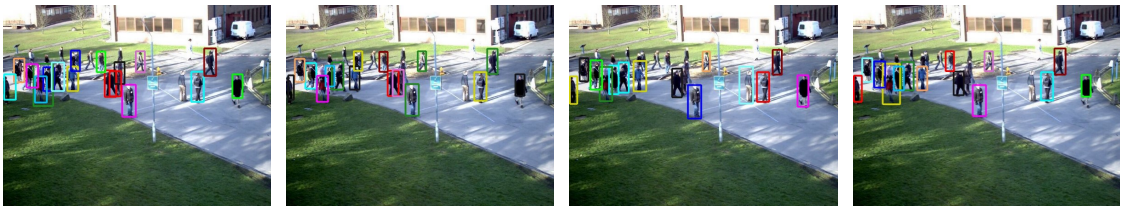
In this section, we compare our visual multi-object tracker with several state-of-the-art methods on the MOT Challenge benchmark [51] in order to show the performance of our algorithm. The synthetic data sets OVVV[52] and virtual KITTI[53] are used as supplementary to the real data in the training. In the single-object tracker, the learning rate for CNN is set to 0.0001 and for fully-connected layers is set to 0.001. In the DRL network, the learning rate λ is set to 0.0001, the regularizer factor ε is set to 0.01, T=20, α=0.95 .

The PETS09-S2L2 sequence consists of 436 frames of 768*576 pixels with heavy crowd density and illumination changes. The pedestrians undergo severe occlusion, scale changes in the sequence. The ADL-Rundle-3 sequence consists of 625 frames of 1920*1080 pixels. It shows a crowded pedestrian street captured from a stationary camera. Frequent occlusions, missed detections, and illumination variation happen among the multiple objects. The TUD-Crossing sequence shows a road crossing from a side view. It consists of 201 frames of 640*480 pixels and includes the non-linear motion, objects in close proximity and occlusions. The AVG-Town Center contains 450 frames of 1920*1080 pixels. It shows a busy town center street from a single elevated camera. The sequence contains medium crowd density, frequent dynamic occlusions and scale changes.

We compare our method (LSTM_DRL) with other state-of-art trackers including RNN-LSTM[50], LP_SSVM[54], MDPSubCNN[55], SiameseCNN[56]. Figure 7, 9, 11, 13 demonstrate qualitative tracking results of our tracker on PETS09-S2L2, ADL-Rundle-3, TUD-Crossing and AVG-Town Center. Figure 8, 10, 12, 14 shows the sample tracking results of other trackers on PETS09-S2L2, ADL-Rundle-3, TUD-Crossing and AVG-Town Center.

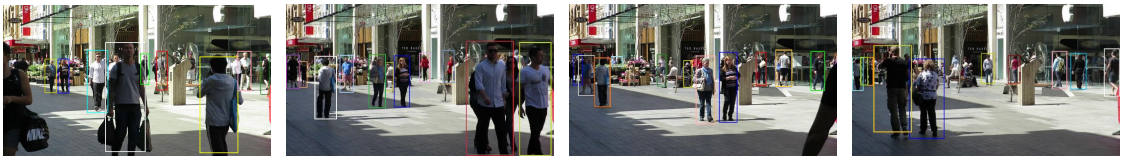
#Frame 15 #Frame 150 #Frame 285 #Frame 420

Figure 7. Qualitative tracking results of our tracker on PETS09-S2L2



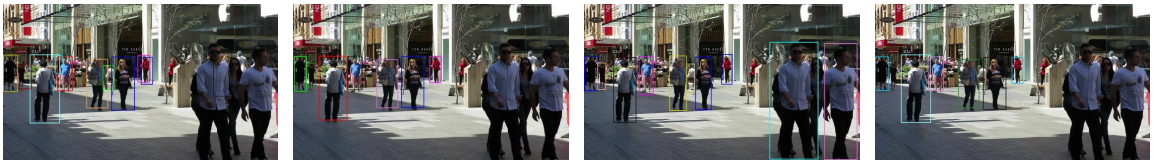
RNN_LSTM LP_SSVM MDP_SubCNN SiameseCNN

Figure 8. Sample tracking results of other trackers on #Frame 15 of PETS09-S2L2



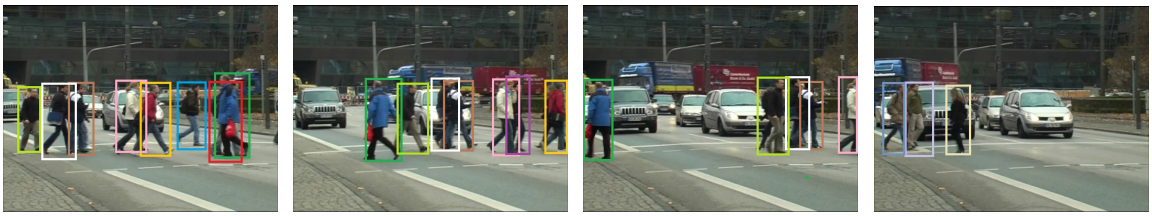
#Frame 30 #Frame 240 #Frame 345 #Frame 600

Figure 9. Qualitative tracking results of our tracker on ADL-Rundle-3



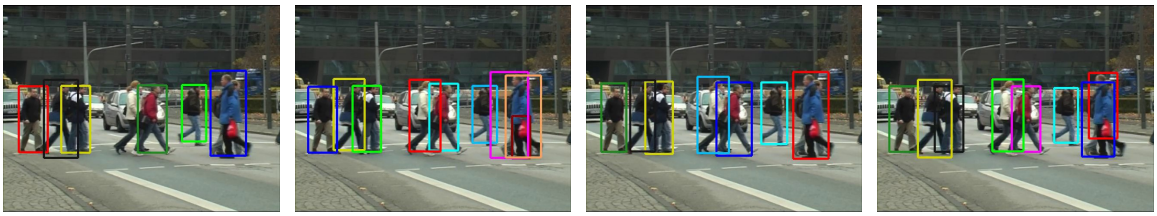
RNN_LSTM LP_SSVM MDP_SubCNN SiameseCNN

Figure 10. Sample tracking results of other trackers on #Frame 240 of ADL-Rundle-3



#Frame 45 #Frame 90 #Frame 120 #Frame 165

Figure 11. Qualitative tracking results of our tracker on TUD-Crossing

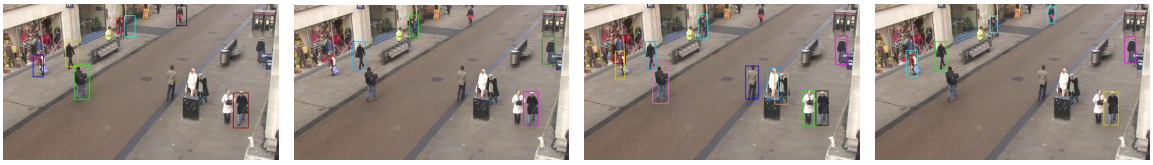


RNN_LSTM LP_SSVM MDP_SubCNN SiameseCNN

Figure 12. Sample tracking results of other trackers on #Frame 45 of TUD-Crossing

#Frame 15 #Frame 135 #Frame 255 #Frame 405

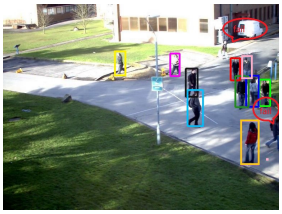
Figure 13. Qualitative tracking results of our tracker on AVG-TownCentre

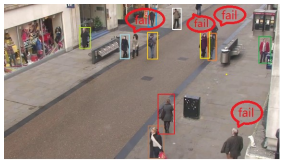


RNN_LSTM LP_SSVM MDP_SubCNN SiameseCNN

Figure 14. Sample tracking results of other trackers on #Frame 405 of AVG-TownCentre

From these experimental results, we can see that our tracker performs well most of time despite frequent occlusions, similarity among objects, scale changes and illumination changes. Nevertheless, there are still some examples of tracking failures unavoidable as illustrated in Figure 15. For example, the brightness of the environment results in the failure of object detection in the #Frame 285 from the PETS09-S2L2 dataset, there are some missing detections in the #Frame 255 from the AVG-TownCentre dataset.





PETS09-S2L2-LSTM-DRL_#Frame 285 AVG-TownCentre- LSTM-DRL _#Frame 255

Figure 15. Selected some failure results of our tracker

To illustrate the contribution of each component, the detection result and the tracking result of single-object trackers are shown in figure 16. Limited to the space, we only list the results on ADL-Rundle-3.

The detection result of YOLO V2 on Frame 30 The tracking result of single-object tracker on Frame 30

Figure 16. The results of detection and single-object tracker on ADL-Rundle-3

From the results, we can see that the object is missed in the detector, while he is tracker in the single-object tracker according DRL.

4.2Quantitative Evaluation

The CLEAR MOT performance metrics are used in this section for quantitative evaluation: the multiple object tracking accuracy (MOTA), the multiple object tracking precision (MOTP), False Positive (FP), and identity switches (IDSW). MOTA evaluates the accuracy composed of false negatives, false positives, and identity switches.

$$\mathrm{MOTA} = 1 - \frac{\sum_t (fn_t + fp_t + \mathrm{IDSW}_t)}{\sum_t gt_t} \tag{12}$$

where fn_t, fp_t, IDSW_t, gt_t are false negatives, false positives, identity switching and ground truth at frame t.

MOTP is the average dissimilarity between all true positives and their corresponding ground truth objects, which calculates the intersection area over the union area of bounding boxes, this is computed as

$$\mathrm{MOTP} = 1 - \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{13}$$

where $d_{t,i}$ denotes the bounding box overlap of object i with its assigned ground truth object, c_t is the number of matches in frame t.

Table 1 reports the quantitative comparison results of our tracker (LSTM_DRL) with other state-of-art trackers on the 11 sequences of MOT Challenge dataset.

Table 1. The quantitative comparison results of our tracker with other state-of-art trackers.

Method	Sequence	MOTA%	MOTP%	FP	IDSW
RNN-LSTM	PETS09-S2L2	38.3	71.6	1016	320
SiameseCNN		47.5	72.6	341	796
MDPSubCNN		34.5	69.7	672	282
LP_SSVM		41.5	70.5	629	212
LSTM_DRL(Our)		45.8	72.9	354	255
RNN-LSTM	ADL-Rundle-3	23.7	72.0	2193	158
SiameseCNN		39.7	72.9	191	33
MDPSubCNN		44.9	79.6	793	56
LP_SSVM		28.0	72.9	1855	81
LSTM_DRL(Ours)		45.2	75.1	651	30
RNN-LSTM	TUD-Crossing	57.2	71.1	81	43
SiameseCNN		73.7	73.0	85	8
MDPSubCNN		78.9	76.7	32	6
LP_SSVM		60.0	74.2	48	18
LSTM_DRL(Ours)		79.1	75.9	30	11
RNN-LSTM	AVG-Town Center	13.4	68.8	1206	299
SiameseCNN		19.3	69.0	698	142

MDPSubCNN		49.5	70.1	1381	121
LP_SSVM		14.7	70.1	459	123
LSTM_DRL(Ours)		38.6	71.0	598	101
RNN-LSTM	Venice-1	12.7	71.7	686	56
SiameseCNN		22.3	73.0	322	4
MDPSubCNN		15.9	72.4	843	47
LP_SSVM		17.8	73.0	696	23
LSTM_DRL(Ours)		23.4	73.8	302	6
RNN-LSTM	ETH-Jelmoli	34.8	73.3	314	59
SiameseCNN		42.3	72.8	315	30
MDPSubCNN		32.9	73.6	639	22
LP_SSVM		39.5	74.4	224	17
LSTM_DRL(Ours)		43.9	75.1	213	13
RNN-LSTM	ETH-Linthescher	12.4	74.7	164	49
SiameseCNN		16.7	74.2	93	27
MDPSubCNN		27.2	74.7	191	48
LP_SSVM		15.6	75.6	41	11
LSTM_DRL(Ours)		27.1	75.4	52	14
eRNN-LSTM	ETH-Crossing	21.1	75.5	27	7
SiameseCNN		27.5	74.1	20	4
MDPSubCNN		28.8	74.7	59	0
LP_SSVM		24.9	75.6	10	2
LSTM_DRL(Ours)		29.6	77.9	12	4
RNN-LSTM	ADL-Rundle-1	-2.2	69.9	4213	241
SiameseCNN		25.6	71.6	1999	33
MDPSubCNN		16.2	71.5	3157	49
LP_SSVM		14.0	71.9	3507	69
LSTM_DRL(Ours)		27.9	72.8	1846	39

From the results of Table 1, we can see that our proposed method provides the highest MOTP values and the lowest FN values on the PETS09-S2L2 dataset, provides the highest MOTA values and the lowest FN and IDSW values on the ADL-Rundle-3 dataset, provides the highest MOTA values and the lowest FP and FN values on the TUD-Crossing dataset, provides the highest MOTP values and the lowest IDSW values on the AVG-Town Center dataset. The proposed method obtains a better performance can mainly be attributed to the three parts of the tracker: YOLO V2 is a state-of-the-art object detector, the data association strategy based on LSTM can find a global optimal assignment, and the single object trackers are able to find the location of the object via deep reinforcement learning.

We implement the experiments of our proposed multi-object tracking algorithm based on Windows 10 operating system and using MATLAB R2016b as the software platform. The configuration of the computer is Intel(R) Core(TM) i7-4712MQ and GeForce GTX TITAN X GPU, 12.00 GB VRAM.

The results of running time on the MOT Challenge test dataset are shown in Table 2, where we compare to some state of-the-art trackers. Our method is a real-time tracking system, although the speed is slower than RNN-LSTM, which does not incorporate appearance, the other performances of our method is better than it.

Table 2. **The running time comparison results of our tracker with other state-of-art trackers.**

Method	**FPS**
RNN-LSTM	166.8
TC-ODAL[8]	2.4
JPDA-m[59]	35.6
MDPSubCNN	2.1
LSTM_DRL (Ours)	108.0

. Conclusion

This paper proposes a visual multi-object tracking algorithm based on LSTM and deep reinforcement learning to overcome the problems of the existing algorithms, such as, they have many limits because hand-crafted features can not capture more complex characteristics of the objects, tracking fails when the number of objects varying, and so on. We adopted the object detector YOLO V2 to detect the multiple objects. The single object tracker is composed of a network that includes a CNN followed by an LSTM unit. Each tracker, regarded as an agent, is trained by utilizing deep reinforcement learning. We conduct a data association using LSTM for each frame between a pre-trained object detector and a number of single object trackers. From the experimental results, we can see that the proposed multi-object tracking method improves the robustness and accuracy of algorithm.

Acknowledgments： This work was supported by the National Key R&D project under grant 2017YFB1002803, the National Natural Science Foundation of China under Grant 61332017 and 61403060, Six Talent Peaks project in Jiangsu Province under Grant 2016XYDXXJS-012, the Natural Science Foundation of Jiangsu Province under Grant BK20171267, 533 talents engineering project in Huaian under Grant HAA201738, Project funded by Jiangsu Overseas Visiting Scholar Program for University Prominent Young & Middle-aged Teachers and Presidents, the Natural Science Foundation of Zhejiang Province under Grant LY15F030010.

Author Contributions: Mingxin Jiang, Chao Deng and Zhigeng Pan conceived and designed the experiments; Xin Chen, Lanfang Wang and Xing Sun performed the experiments; Mingxin Jiang wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ak, K. C.; Jacques, L.; De, V. C. Discriminative and efficient label propagation on complementary graphs for multi-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 39, 61-74.
2. Jiang, M. X.; Pan, Z. G.; Tang, Z. Z. Visual Object Tracking Based on Cross-Modality Gaussian-Bernoulli Deep Boltzmann Machines with RGB-D Sensors. *Sensors*, 2017, 17,121-138.
3. Naiel, M. A.; Ahmad, M. O.; Swamy, M. N. S., et al. Online multi-object tracking via robust collaborative model and sample selection. *Computer Vision & Image Understanding*, 2017, 154:94-107.
4. Shitrit H B, Berclaz J, Fleuret F, et al. Multi-Commodity Network Flow for Tracking Multiple People[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 36(8):1614-27.
5. Milan, A.; Roth, S.; Schindler, K. Continuous Energy Minimization for Multitarget Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, 36:58-72.
6. Bae, S. H.; Yoon, K. J. Robust Online Multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014:1218-1225.
7. Andriyenko, A.; Schindler, K. Multi-target tracking by continuous energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, 38:2054-2066.
8. Girshick R. Fast R-CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015:1440-1448.
9. Ren, S.; He, K.; Girshick, R., et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, 39: 1137 - 1149.
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016:779-788.
11. Brendel, W.; Amer, M.; Todorovic, S. Multiobject tracking as maximum weight independent set. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011:1273-1280.
12. Henriques, J. F.; Caseiro, R.; Batista, J. Globally optimal solution to multi-object tracking with merged measurements. In Proceedings of the IEEE Conference on Computer Vision (ICCV). 2011:2470-2477.

13. Butt, A. A.; Collins, R. T. Multi-target Tracking by Lagrangian Relaxation to Min-cost Network Flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013:1846-1853.
14. Thangali A. Coupling detection and data association for multiple object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012:1948-1955.
15. Bae, S. H.; Yoon, K. J. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017.
16. Yang, E.; Gwak, J.; Jeon, M. Conditional Random Field (CRF)-Boosting: Constructing a Robust Online Hybrid Boosting Multiple Object Tracker Facilitated by CRF Learning. *Sensors*, 2017, 17(3):617.
17. Breitenstein, M. D.; Reichlin, F.; Leibe, B. et al. Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera. *IEEE Trans. Pattern Anal. Mach. Intell.* 2011, 33(9):1820.
18. Dehghan, A.; Tian, Y.; Torr, P. H. S., et al. Target Identity-aware Network Flow for online multiple target tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015:1146-1154.
19. Wen, L.; Lei, Z.; Lyu, S., et al. Exploiting Hierarchical Dense Structures on Hypergraphs for Multi-Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 2016, 38(10):1983-1996.
20. He, Z.; Li, X.; You, X., et al. Connected Component Model for Multi-Object Tracking. *IEEE Trans. Image Processing*, 2016, 25(8):3698-3711.
21. Zhao S, Liu Y, Han Y, et al. Pooling the Convolutional Layers in Deep ConvNets for Video Action Recognition[J]. IEEE Trans. Circuits & Systems for Video Technology, 2017, DOI: 10.1109/TCSVT.2017.2682196.
22. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition[C], CVPR, IEEE,2015:1110-1118.
23. Cao, C.; Xu, W.; Ramanan, D.; et al. Look and Think Twice: Capturing Top-Down Visual Attention with Feedback Convolutional Neural Networks[C], CVPR, IEEE, 2016:2956-2964.
24. Alahi, A.; Goel, K.; Ramanathan, V., et al. Social LSTM: Human Trajectory Prediction in Crowded Spaces[C], CVPR, IEEE, 2016:961-971.
25. Navneet, D.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–26 June 2005.
26. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.***2004**, *57*, 137–154.
27. Wang, X.Y.; Han, T.X.; Yan, S.C. An HOG-LBP human detector with partial occlusion handling. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009.
28. Felzenszwalb, P.; Girshick, R.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645.
29. Lu, J.W.; Wang, G.; Deng, W.H.; Moulin, P.; Zhou, J. Multi-manifold deep metric learning for image set classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
30. Donahue, J. et al. DeCAF: A deep convolutional activation feature for generic visual recognition. In Proc. Int. Conf. Mach. Learn., 2014.
31. Li, H.X.; Li, Y.; Porikli, F. DeepTrack: Learning Discriminative Feature Representations Online for Robust Visual Tracking. *IEEE Trans. Image Process.* **2016**, *25*, 1834–1848.
32. Wang, N.Y.; Yeung, D.Y. Learning a deep compact image representation for visual tracking. In Proceedings of the Conference and Workshop on Neural Information Processing Systems (NIPS), South Lake Tahoe, NV, USA, 5–10 December 2013.
33. Wang, L.; Liu, T.; Wang, G.; Chan, K.L.; Yang, Q.X. Video Tracking Using Learned Hierarchical Features. *IEEE Trans. Image Process.* **2015**, *24*, 1424–1435.
34. Fan, J.; Xu, W. ; Wu, Y. and Gong, Y. Human tracking using convolutional neural networks. *IEEE Trans. Neural Netw.* **2010,** 21, 1610–1623.
35. Mnih, V.; Kavukcuoglu, K.; Silver, D. ; Graves, A.; MilanMiloglou, I.; Wierstra, D. and Riedmiller M. Playing atari with deep reinforcement learning[J]. arXiv:1312.5602, 2013.
36. Silver, D. ; Huang, A.; Maddison, C. J. ; Guez, A.; Sifre, L. ; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V. ; Lanctot, M. et al. Mastering the game of go with deep neural networks and tree search[J]. Nature, 529(7587):484–489, 2016.

37. Jayaraman, D.; Grauman, K. Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion. arXiv:1605.00164, 2016.
38. Caicedo, J. C.; Lazebnik. S. Active object localization with deep reinforcement learning. CVPR, 2015: 2488–2496.
39. Zhang, D.; Maei, H.; Wang, X., et al. Deep Reinforcement Learning for Visual Object Tracking in Videos. *arXiv preprint*, 2017.
40. Jie, Z.; Liang, X.; Feng, J. et al. Tree-Structured Reinforcement Learning for Sequential Object Localization. In Advances in Neural Information Processing, 2016:127-135.
41. Luo, W.; Sun, P.; Mu, Y., et al. End-to-end Active Object Tracking via Reinforcement Learning. *arXiv* preprint, 2017.
42. Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; Choi, J. Y. Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning, CVPR 2017.
43. Hochreiter, S.; Schmidhuber, J. Long short- term memory. Neural computation. 1997,9, 1735–1780.
44. Graves, A. Generating Sequences With Recurrent Neural Networks[J]. Computer Science, 2013, arXiv: 1308. 0850.
45. Wang, Z. ; Freitas, N.; Lanctot, M. Dueling network architectures for deep reinforcement learning. arXiv: 1511.06581, 2015.
46. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. CoRR, abs/1509.06461, 2015.
47. Peters, J. Policy gradient methods[J], Scholarpedia, 2010, 5(11):3698.
48. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. CVPR 2017.
49. Milan, A.; Rezatofighi, S. H.; Dick, A. et al. Online Multi-Target Tracking Using Recurrent Neural Networks. AAAI 2017.
50. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S. & Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. arXiv:1504.01942.
51. Taylor G R, Chosak A J, Brewer P C. OVVV: Using Virtual Worlds to Design and Evaluate Surveillance Systems[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007:1-8.
52. Gaidon A, Wang Q, Cabon Y, et al. VirtualWorlds as Proxy for Multi-object Tracking Analysis[C]// Computer Vision and Pattern Recognition. IEEE, 2016:4340-4349.
53. Wang, S.; Fowlkes, C. Learning Optimal Parameters for Multi-target Tracking with Contextual Interactions. In International Journal of Computer Vision, 2016.
54. Xiang, Y.; Alahi, A.; Savarese, S. Learning to Track: Online Multi-Object Tracking by Decision Making. In International Conference on Computer Vision (ICCV), pp. 4705-4713, 2015.
55. Leal-Taixé, L.; Canton-Ferrer, C.; Schindler, K. Learning by Tracking: Siamese CNN for Robust Target Association. DeepVision Workshop (CVPR), Las Vegas (Nevada, USA), June 2016.
56. Rezatofighi S H, Milan A, Zhang Z, et al. Joint Probabilistic Data Association Revisited[C]// IEEE International Conference on Computer Vision. IEEE, 2015:3047-3055.