Hindawi
Complexity
Volume 2018, Article ID 4695890, 12 pages
https://doi.org/10.1155/2018/4695890

Research Article

Multiobject Tracking in Videos Based on LSTM and Deep Reinforcement Learning

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

[1]*Faculty of Electronic Information Engineering, Huaiyin Institute of Technology, Huaian 223003, China*
[2]*School of Physics & Electronic Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China*
[3]*Institute of Industrial VR, Foshan University, Foshan 528000, China*
[4]*Digital Media & Interaction Research Center, Hangzhou Normal University, Hangzhou 310012, China*
[5]*Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian 223003, China*

Correspondence should be addressed to Chao Deng; dengchao_hpu@163.com and Zhi-geng Pan; zgpan@hznu.edu.cn

Received 13 September 2017; Revised 20 March 2018; Accepted 28 March 2018; Published 19 November 2018

Academic Editor: Michele Scarpiniti

Multiple-object tracking is a challenging issue in the computer vision community. In this paper, we propose a multiobject 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 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 the object 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-the-art methods. Multiple targets can be steadily tracked even when frequent occlusions, similar appearances, and scale changes happened.

1. Introduction

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

1.1. Related Work. 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 the recent tracking-by-detection algorithms aim at decomposing multiobject 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 multiobject tracking method can recognize emerging or disappearing objects in the sequences of a video more easily, and the search space of object 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 multiobject 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. Bae and Yoon [15]

proposed a novel online visual multiobject 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 multihypotheses 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 multiobject tracking algorithms have concentrated on enhancing the performance of the object detector or designing 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], and human trajectory prediction [25]. The main advantages of LSTMs for modeling sequential data is that they allow end-to-end fine-tuning 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 nonlinear transformations and memory cells, which makes it effective for data association.

Most previous multiobject tracking methods represent objects using raw pixel and low-level handcrafted features, 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 handcrafted features cannot capture more complex characteristics of the objects. Recently, deep learning has received much attention with state-of-the-art results in complicated tasks such as object detection [29], image classification [30], object recognition [31], and object tracking [32]. A deep-learning tracker (DLT) was proposed in [33], which uses a stacked denoising autoencoder to learn the generic features from a large number of auxiliary images offline. However, the DLT tracker cannot describe the temporal invariance of deep features, which is important for visual object tracking. In [34], a deep-learning tracking method was developed 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 the recent exciting achievements of deep learning, integrating deep-learning methods with RL has recently shown very promising results on decision-making problems, that is, deep reinforcement learning (DRL). Deep neural networks are able to make reinforcement learning algorithms perform more effectively because they can provide deep feature representations. DRL algorithms have achieved unequalled success in many challenging domains, for example, Atari games [36] and 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]. Yun 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 action-decision networks for visual tracking with deep reinforcement learning. However, these tracking methods based on deep reinforcement learning usually focus on a single object; there is little work related to multiobject tracking. Unlike the aforementioned methods, our method exploits how to apply deep reinforcement learning to solve the online multiobject tracking problem.

1.2. Summary of Contributions. Our motivation is to design a real-time multiple-object tracker via LSTM and DRL, which can incorporate 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:

(i) We propose a novel visual multiobject 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.

(ii) The proposed multiobject tracker includes three modules: an object detection module, a number of single-object trackers, and a data association module. We adopt YOLO V2 as an object detector as it is a real-time detection system. Each single-object tracker is treated as an agent, which is trained using DRL. An LSTM-based architecture is adopted to solve the joint data association problem.

(iii) To compare our multiobject 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: Section 2 reviews the background. Section 3 introduces the proposed multiobject 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

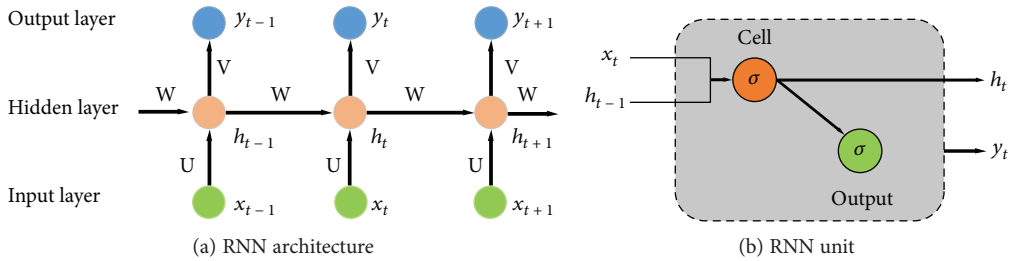


Figure 1: Recurrent neural networks.

dynamics, as shown in Figure 1. The formulas that govern the computation happening in a RNN are as follows:

$$\begin{aligned} \mathbf{h}_t &= f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}), \\ \mathbf{y}_t &= \text{soft max}(\mathbf{V}\mathbf{h}_t), \end{aligned} \tag{1}$$

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

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 problem of exploding and vanishing gradients. 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.

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

$$\begin{aligned} i_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + b_i), \\ f_t &= \sigma\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + b_f\right), \\ \mathbf{c}_t &= f_t \odot \mathbf{c}_{t-1} + i_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + b_c), \\ o_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + b_o), \\ \mathbf{h}_t &= o_t \odot \tanh(\mathbf{c}_t), \end{aligned} \tag{2}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic sigmoid function, $\mathbf{c}_t$ is the cell input activation vector, i_t describes the input gate, f_t represents the forget gate, and o_t output gate. All of the above are the same size as the hidden vector $\mathbf{h}_t$. That is, in addition to a hidden vector $\mathbf{h}_t \in \mathbb{R}^N$, 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 $\mathbf{c}_t \in \mathbb{R}^N$. We can find the meaning of the weight matrix; for example, $\mathbf{W}_{hi}$ represents the hidden to input gate matrix and $\mathbf{W}_{xo}$ represents the input to output gate matrix. b_i, b_f, b_o, and b_c are the bias terms which are added to i, f, o, and c.

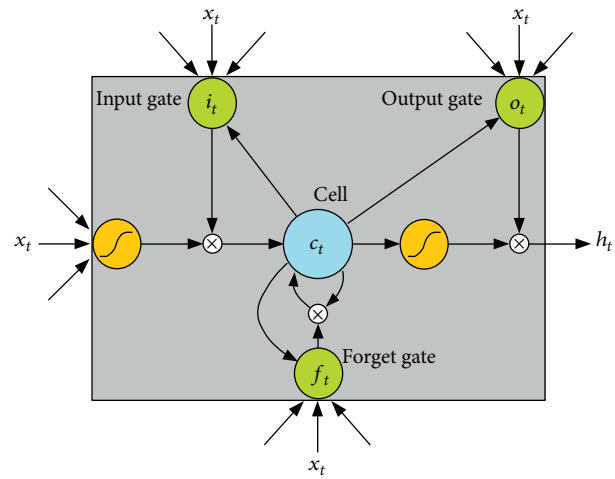


Figure 2: Long short-term memory unit.

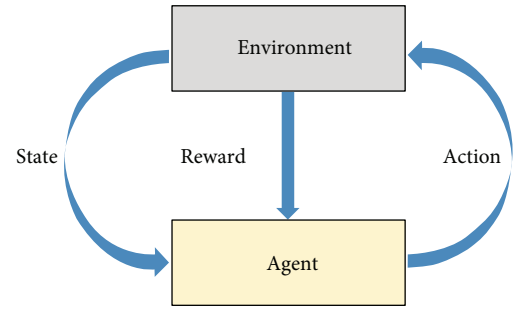


Figure 3: The typical framing of a reinforcement learning.

2.2. Deep Reinforcement Learning (DRL). Reinforcement learning (RL) can usually be used to solve sequential decision-making problems. The process of reinforcement learning is 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 are

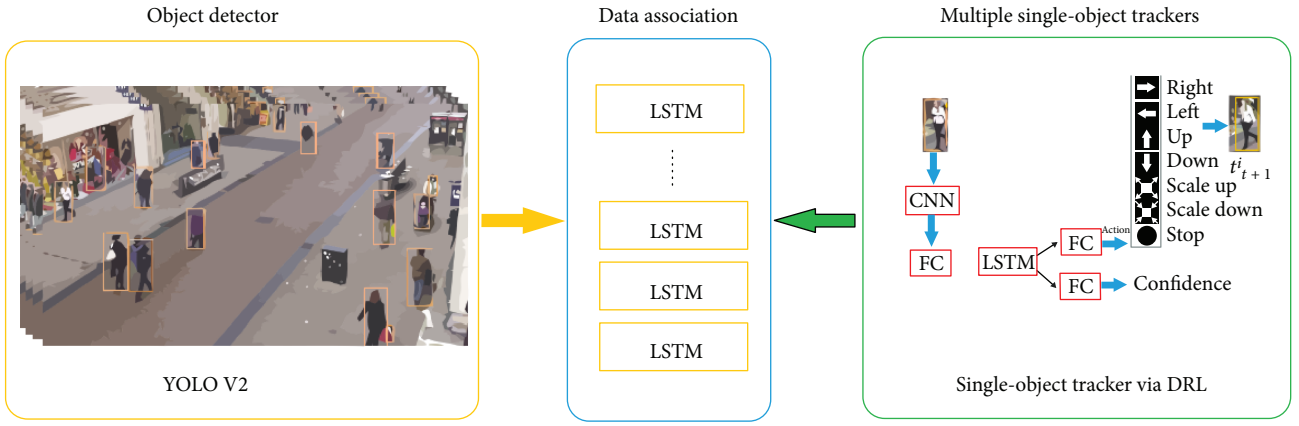


Figure 4: Overview of our proposed multiobject tracking algorithm.

based on the DQN algorithm such as dueling DQN [46] and double DQN [47].

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 and they do not suffer from the difficult problem caused by uncertain state information.

3. Proposed Visual Multiobject Tracking Algorithm

In Subsections 3.1–3.3, we show a brief architecture of our proposed multiobject tracking algorithm firstly. The details of our method are described in the following content.

3.1. Architecture of the Proposed Multiobject 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 an 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 intersection-over-union (IoU) distance is computed between the ground truth and the detections at the first frame. The IoU distance between the mean of its short-term history of validated detections and the current detections is also computed to obtain the correct detections at the other frame. Secondly, 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 an LSTM-based architecture that can learn to solve the joint data association problem from training data.

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 of 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 in Figure 5. This section presents details of these components.

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 subsets: horizontal moves {right, left}, vertical moves {up, down}, and scale changes {scale up, scale down}.

The state definition is a tuple $s_t = (\mathbf{p}_t, \mathbf{v}_t)$, where $\mathbf{p}_t$ is the image patch (which is pointed by a 4-dimensional vector $\mathbf{p}_t = [\mathbf{x}_t, \mathbf{y}_t, \mathbf{w}_t, \mathbf{h}_t]$) within the bounding box of the object and $\mathbf{v}_t$ is a vector with the history of taken actions. The history vector stores the past 10 actions, which means $\mathbf{v}_t$ has 70 dimensions as each action vector has 7 dimensions. At time step $t+1$, the state $s_{t+1} = (\mathbf{p}_{t+1}, \mathbf{v}_{t+1})$ is decided by $s_t = (\mathbf{p}_t, \mathbf{v}_t)$ and the state transition functions, where $\mathbf{p}_{t+1} = f_t(\mathbf{p}_t, a_t)$ and $\mathbf{v}_{t+1} = f_v(\mathbf{v}_t, a_t)$.

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 the "stop" action is selected at termination step T, the reward signal r_T is a thresholding function of IoU as follows:

$$r_T = 1, \quad \text{if } \mathrm{IoU}(\mathbf{p}_T, \mathrm{g}) > \tau,$$
$$r_T = -1, \quad \textit{otherwise}, \tag{3}$$

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

We adopt policy-based reinforcement learning methods as they have a better capability of learning random policies and convergence properties. Our whole network is

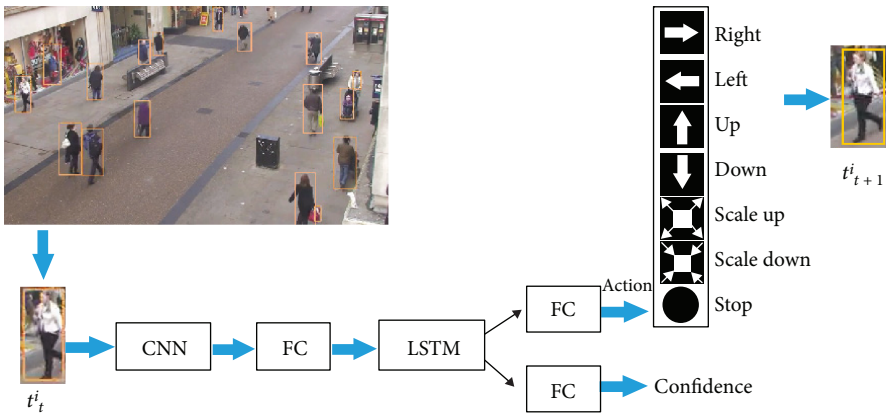


Figure 5: The pipeline of the single-object tracker.

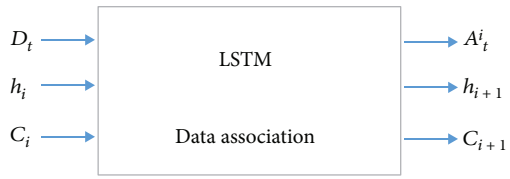


Figure 6: LSTM-based architecture for data association.

parameterized by W, the policy-based method models, the policy function $\pi(a \mid s; W)$, and the value function $V(a \mid s; W')$; 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 $\pi(a_t \mid s_t; W)$. Approximation of the policy function can be obtained by a stochastic gradient ascent algorithm. As there are very limited amounts of labelled data for multiobject tracking, we use synthetic data as a supplementary to the real data in the training. The parameters W and W' can be learned according to the following equations:

$$\begin{aligned} W &\leftarrow W + \lambda\left(R_t - V\left(a_t|s_t; W'\right)\right)\nabla_W \log \pi(a_t|s_t; W) \\ &\quad + \varepsilon\nabla_W H(\pi(\cdot|s_t; W)), \\ W' &\leftarrow W' - \lambda\left(R_t - V\left(a_t|s_t; W'\right)\right), \end{aligned} \tag{4}$$

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, $0 < \alpha \le 1$, λ is the learning rate, $H(\cdot)$ is an entropy regularizer, and ε is the regularizer factor.

Our deep CNN is conducted on the VGG-16 network, which includes five pooling stages, that is, Conv1-2, Conv2-2, Conv3-3, Conv4-3, and 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 a size $M/2^5 \times N/2^5$. 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 $\mathbf{P}_t = \{\mathbf{p}_t^i\}_{i=1}^M$ represent the set of all outputs of single-object trackers at time step t, $\mathbf{p}_t^i$ refers to the state of the ith output of a single-object tracker, and M is the number of objects that can be tracked simultaneously in one time step. The state of the ith object is represented by the 4-dimensional vector $\mathbf{p}_t^i = [\mathbf{x}_t^i, \mathbf{y}_t^i, \mathbf{w}_t^i, \mathbf{h}_t^i]$. We define $Q_t = \{q_t^j\}_{j=1}^N$ as the set of detections from the object detector with q_t^j the jth detection and N the number of detections. Let $\mathbf{D}_t \in \mathbb{R}^{M\times N}$ denote the similarity matrix for data association that measures the relation between an output of single-object tracker $\mathbf{p}_t^i$ and a detection m_t^j, where $D_t^{ij} = \|\mathbf{p}_t^i - q_t^j\|_2$ is the Euclidean distance between $\mathbf{p}_t^i$ and q_t^j. Data association based on LSTM for object i is illustrated in Figure 6.

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 $\mathbf{D}_t$. The output are the hidden state h_{i+1}, the cell state c_{i+1}, and the assignment probability vector $\mathbf{A}_t^i$. $\mathbf{A}_t^i$ 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. $\mathbf{A}_t^{ij} = a$ (object i assigned to the jth detection) and $\sum_j \mathbf{A}_t^{ij} = 1$. Let ε be the correct assignment; we adapt the negative log-likelihood loss as the cost function to measure the misassignment cost:

$$C\left(\mathbf{A}_t^i, \varepsilon\right) = -\log\left(\mathbf{A}_t^{i\varepsilon}\right), \tag{5}$$

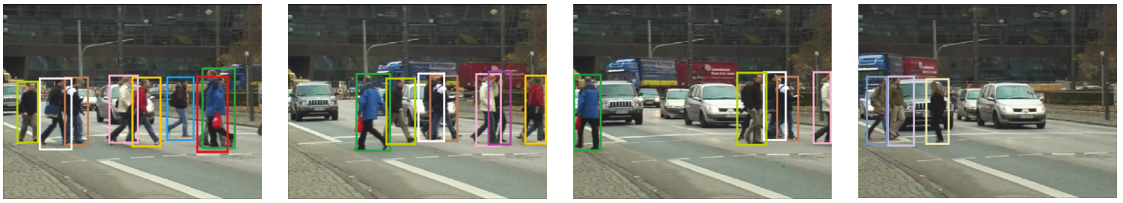
The data association requires more representation power, so it is a more complex task. The data-association-module-based LSTM include two layers and 512 hidden units. It takes approximately 40 hours to train all the modules in our tracker on a CPU. The training can be sped up significantly by using GPUs.

#Frame 15 (a) #Frame 150 (b) #Frame 285 (c) #Frame 420 (d)

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

#Frame 30 (a) #Frame 240 (b) #Frame 345 (c) #Frame 600 (d)

Figure 8: Qualitative tracking results of our tracker on ADL-Rundle-3.



#Frame 45 (a) #Frame 90 (b) #Frame 120 (c) #Frame 165 (d)

Figure 9: Qualitative tracking results of our tracker on TUD-Crossing.

4. Experiments

4.1. Qualitative Evaluation. In this section, we compare our visual multiobject 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 datasets 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 it is set to 0.001. In the DRL network, the learning rate λ is set to 0.0001, and the regularizer factor ε is set to 0.01, $T = 20$, $\alpha = 0.95$.

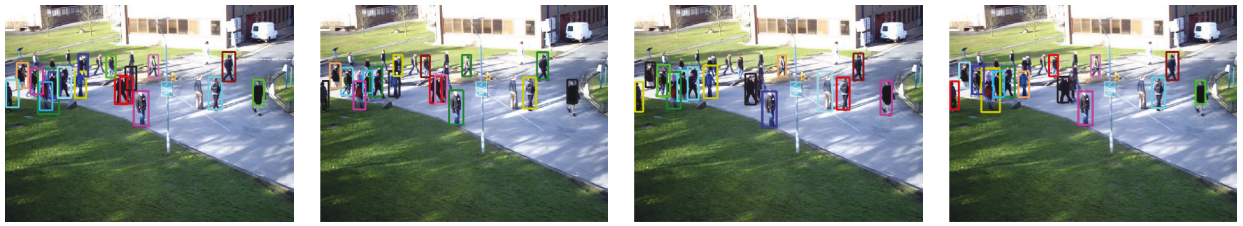
The PETS09-S2L2 sequence consists of 436 frames of 768*576 pixels with heavy crowd density and illumination changes. The pedestrians undergo severe occlusion and scale changes in the sequence. The ADL-Rundle-3 sequence consists of 625 frames of 1920*1080 pixels. It shows a crowded pedestrian street captured 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 nonlinear 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-the-art trackers including RNN-LSTM [50], LP_SSVM [54], MDPSubCNN [55], and SiameseCNN [56]. Figures 7, 8, 9, and 10 demonstrate the qualitative tracking results of our tracker on PETS09-S2L2, ADL-Rundle-3, TUD-Crossing, and AVG-Town Center. Figures 11, 12, 13, and 14 show the sample tracking results of other trackers on PETS09-S2L2, ADL-Rundle-3, TUD-Crossing, and AVG-Town Center.

From these experimental results, we can see that our tracker performs well most of the time despite frequent occlusions, similarity among objects, scale changes, and illumination changes. Nevertheless, there are still some examples of unavoidable tracking failures as illustrated in Figure 15. For example, the brightness of the environment

#Frame 15 (a) #Frame 135 (b) #Frame 255 (c) #Frame 405 (d)

Figure 10: Qualitative tracking results of our tracker on AVG-TownCentre.

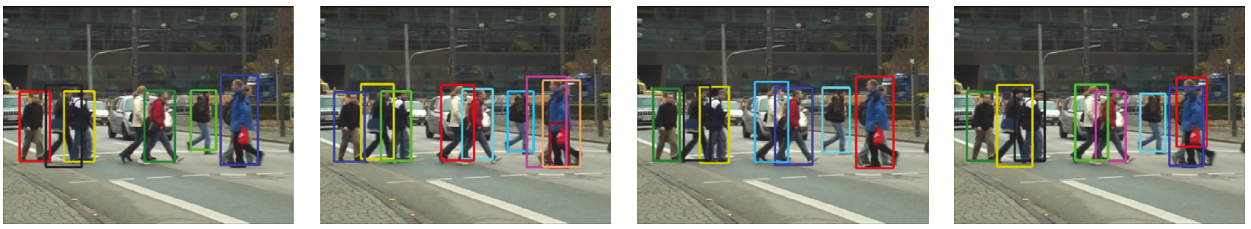


RNN_LSTM (a) LP_SSVM (b) MDP_SubCNN (c) SiameseCNN (d)

Figure 11: Sample tracking results of other trackers on frame number 15 of PETS09-S2L2.

RNN_LSTM (a) LP_SSVM (b) MDP_SubCNN (c) SiameseCNN (d)

Figure 12: Sample tracking results of other trackers on frame number 240 of ADL-Rundle-3.



RNN_LSTM (a) LP_SSVM (b) MDP_SubCNN (c) SiameseCNN (d)

Figure 13: Sample tracking results of other trackers on frame number 45 of TUD-Crossing.

results in the failure of object detection in frame number 285 from the PETS09-S2L2 dataset and there are some missing detections in frame number 255 from the AVG-TownCentre dataset.

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.

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

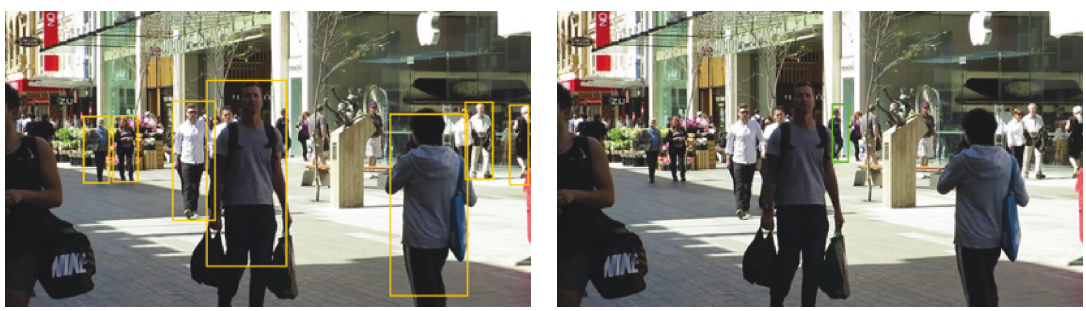
4.2. Quantitative Evaluation. The CLEAR MOT performance metrics are used in this section for quantitative evaluation: the multiple-object tracking accuracy (MOTA), the multiple-object tracking precision (MOTP), false positive

RNN_LSTM (a) LP_SSVM (b) MDP_SubCNN (c) SiameseCNN (d)

Figure 14: Sample tracking results of other trackers on frame number 405 of AVG-TownCentre.

(a) PETS09-S2L2-LSTM-DRL frame number 285 (b) AVG-TownCentre-LSTM-DRL frame number 255

Figure 15: Some selected failure results of our tracker.



(a) The detection result of YOLO V2 on frame number 30 (b) The tracking result of the single-object tracker on frame number 30

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

(FP), and identity switches (IDSW). MOTA evaluates the accuracy composed of false negatives, false positives, and identity switches.

$$\text{MOTA} = 1 - \frac{\sum_t (fn_t + fp_t + \text{IDSW}_t)}{\sum_t gt_t}, \quad (6)$$

where fn_t, fp_t, IDSW_t, and 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

$$\text{MOTP} = 1 - \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \quad (7)$$

where $d_{t,i}$ denotes the bounding box overlap of object i with its assigned ground truth object and 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-the-art trackers on the 11 sequences of the MOT Challenge dataset.

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, and provides the highest MOTP values and the lowest IDSW values on the AVG-Town Center dataset. The proposed method obtains a better performance that 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 multiobject tracking algorithm based on the Windows 10

Table 1: The quantitative comparison results of our tracker with other state-of-the-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 (ours)		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

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

Method	FPS
RNN-LSTM	166.8
TC-ODAL [8]	2.4
JPDA-m [56]	35.6
MDPSubCNN	2.1
LSTM_DRL (ours)	108.0

operating system and using MATLAB R2016b as the software platform. The configuration of the computer is Intel® Core™ 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 they are compared to some state-of-the-art trackers. Our method is a real-time tracking system and although the speed is slower than RNN-LSTM, which does not incorporate appearance, the other performance of our method is better than it.

5. Conclusion

This paper proposes a visual multiobject tracking algorithm based on LSTM and deep reinforcement learning to overcome the problems of the existing algorithms: they have many limits because handcrafted features cannot capture more complex characteristics of the objects, tracking fails when the number of objects vary, 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 data association using LSTM for each frame between a pretrained object detector and a number of single-object trackers. From the experimental results, we can see that the proposed multiobject tracking method improves the robustness and accuracy of the algorithm.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Ming-xin Jiang, Chao Deng, and Zhi-geng Pan conceived and designed the experiments; Lan-fang Wang, and Xing Sun performed the experiments; and Ming-xin Jiang wrote the paper.

Acknowledgments

This work was supported by the National Key R&D Project under Grant no. 2017YFB1002803, the National Natural Science Foundation of China under Grant no. 61332017, the Six Talent Peaks Project in Jiangsu Province under Grant no. 2016XYDXXJS-012, the Natural Science Foundation of Jiangsu Province under Grant no. BK20171267, the 533 Talents Engineering Project in Huaian under Grant no. HAA201738, and a project funded by the Jiangsu Overseas Visiting Scholar Program for University Prominent Young & Middle-Aged Teachers and Presidents. This work also received support from the Major Program of the Natural Science Research of Jiangsu Higher Education Institutions of China (18KJA520002), a project funded by the Jiangsu Laboratory of Lake Environment Remote Sensing Technologies (JSLERS-2018-005), and the fifth issue 333 high-level talent training project of the Government of Jiangsu Province (BRA2018333).

References

[1] K. C. Amit Kumar, L. Jacques, and C. de Vleeschouwer, "Discriminative and efficient label propagation on complementary graphs for multi-object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 61–74, 2017.

[2] M. Jiang, Z. Pan, and Z. Tang, "Visual object tracking based on cross-modality Gaussian-Bernoulli deep Boltzmann machines with RGB-D sensors," *Sensors*, vol. 17, no. 1, p. 121, 2017.

[3] M. A. Naiel, M. O. Ahmad, M. N. S. Swamy, J. Lim, and M. H. Yang, "Online multi-object tracking via robust collaborative model and sample selection," *Computer Vision and Image Understanding*, vol. 154, pp. 94–107, 2017.

[4] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1614–1627, 2014.

[5] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, 2014.

[6] S. H. Bae and K. J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1218–1225, Columbus, OH, USA, June 2014.

[7] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, June 2011.

[8] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1440–1448, Santiago, Chile, December 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[11] W. Brendel, M. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1273–1280, Providence, RI, USA, June 2011.

[12] J. F. Henriques, R. Caseiro, and J. Batista, "Globally optimal solution to multi-object tracking with merged measurements," in *Proceedings of the IEEE Conference on Computer Vision (ICCV)*, pp. 2470–2477, Barcelona, Spain, November 2011.

[13] A. A. Butt and R. T. Collins, "Multi-target tracking by Lagrangian relaxation to min-cost network flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1846–1853, Portland, OR, USA, June 2013.

[14] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1948–1955, Providence, RI, USA, June 2012.

[15] 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. 40, no. 3, pp. 595–610, 2018.

[16] E. Yang, J. Gwak, and M. Jeon, "Conditional random field (CRF)-boosting: constructing a robust online hybrid boosting multiple object tracker facilitated by CRF learning," *Sensors*, vol. 17, no. 3, p. 617, 2017.

[17] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1820–1833, 2011.

[18] A. Dehghan, Y. Tian, P. H. S. Torr, and M. Shah, "Target identity-aware network flow for online multiple target tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1146–1154, Boston, MA, USA, June 2015.

[19] L. Wen, Z. Lei, S. Lyu, S. Z. Li, and M. H. Yang, "Exploiting hierarchical dense structures on hypergraphs for multi-object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 1983–1996, 2016.

[20] Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang, "Connected component model for multi-object tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3698–3711, 2016.

[21] S. Zhao, Y. Liu, Y. Han, R. Hong, Q. Hu, and Q. Tian, "Pooling the convolutional layers in deep ConvNets for video action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, 2018.

[22] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1110–1118, Boston, MA, USA, June 2015.

[23] C. Cao, X. Liu, Y. Yang et al., "Look and think twice: capturing top-down visual attention with feedback convolutional neural networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2956–2964, Santiago, Chile, December 2016.

[24] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, Las Vegas, NV, USA, June 2016.

[25] D. Navneet and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, June 2005.

[26] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[27] X. Y. Wang, T. X. Han, and S. C. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[28] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[29] J. W. Lu, G. Wang, W. H. Deng, P. Moulin, and J. Zhou, "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, June 2015.

[30] J. Donahue, Y. Jia, O. Vinyals et al., "DeCAF: a deep convolutional activation feature for generic visual recognition," in *Proceedings of International Conference on Machine Learning*, pp. 647–655, Bejing, China, June 2014.

[31] H. Li, Y. Li, and F. Porikli, "DeepTrack: learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2016.

[32] N. Y. Wang and D. Y. Yeung, "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, December 2013.

[33] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1424–1435, 2015.

[34] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010.

[35] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning," 2013, http://arxiv.org/abs/1312.5602.

[36] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[37] D. Jayaraman and K. Grauman, "Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion," 2016, http://arxiv.org/abs/1605.00164.

[38] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2488–2496, Santiago, Chile, December 2015.

[39] D. Zhang, H. Maei, X. Wang, and Y.-F. Wang, "Deep reinforcement learning for visual object tracking in videos," 2017, http://arxiv.org/abs/1701.08936.

[40] Z. Jie, X. Liang, J. Feng, X. Jin, W. F. Lu, and S. Yan, "Tree-structured reinforcement learning for sequential object localization," in *Conference on Neural Information Processing Systems*, pp. 127–135, Barcelona, Spain, 2016.

[41] W. Luo, P. Sun, Y. Mu, and W. Liu, "End-to-end active object tracking via reinforcement learning," 2017, http://arxiv.org/abs/1705.10561.

[42] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] A. Graves, "Generating sequences with recurrent neural networks," http://arxiv.org/abs/1308.0850.

[45] Z. Wang, N. Freitas, and M. Lanctot, "Dueling network architectures for deep reinforcement learning," 2015, http://arxiv.org/abs/1511.06581.

[46] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," http://arxiv.org/abs/1509.06461.

[47] J. Peters, "Policy gradient methods," *Scholarpedia*, vol. 5, no. 11, p. 3698, 2010.

[48] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2016.

[49] A. Milan, S. H. Rezatofighi, A. Dick, I. D. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," http://arxiv.org/abs/1604.03635.

[50] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: towards a benchmark for multi-target tracking," http://arxiv.org/abs/1504.01942.

[51] G. R. Taylor, A. J. Chosak, and P. C. Brewer, "OVVV: using virtual worlds to design and evaluate surveillance systems," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[52] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "VirtualWorlds as proxy for multi-object tracking analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4340–4349, Las Vegas, NV, USA, June 2016.

[53] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 484–501, 2017.

[54] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: online multi-object tracking by decision making," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4705–4713, Santiago, Chile, December 2015.

[55] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Las Vegas, NV, USA, 2016.

[56] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *IEEE International Conference on Computer Vision*, pp. 3047–3055, Santiago, Chile, December 2015.