IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 14, 2019, accepted February 12, 2019, date of publication February 26, 2019, date of current version March 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901300

Multi-Agent Deep Reinforcement Learning for Multi-Object Tracker

MINGXIN JIANG[1], TAO HAI[2], ZHIGENG PAN[3], HAIYAN WANG[1], YINJIE JIA[1], AND CHAO DENG[4]

[1]Jiangsu Laboratory of Lake Environment Remote Sensing Technologies, Huaiyin Institute of Technology, Huaian 223003, China
[2]Computer Science Department, Baoji University of Arts and Sciences, Baoji 721031, China
[3]Digital Media and Interaction Research Center, Hangzhou Normal University, Hangzhou 310012, China
[4]School of Physics and Electronic Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China

Corresponding authors: Mingxin Jiang (jiangmingxin@126.com) and Zhigeng Pan (zgpan@hznu.edu.cn)

This work was supported in part by the National Key Research and Development project under Grant 2017YFB1002803, in part by the Major Program of Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant 18KJA520002, in part by the Jiangsu Laboratory of Lake Environment Remote Sensing Technologies under Grant JSLERS-2018-005, in part by Six talent peaks project in Jiangsu Province under Grant 2016XYDXXJS-012, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20171267, in part by the fifth issue 333 high-level talent training project of Jiangsu province under Grant BRA2018333, in part by the 533 Talents Engineering Project in Huaian under Grant HAA201738, and in part by the National Natural Science Foundation of China under Grant 61801188.

ABSTRACT Multi-object tracking has been a key research subject in many computer vision applications. We propose a novel approach based on multi-agent deep reinforcement learning (MADRL) for multi-object tracking to solve the problems in the existing tracking methods, such as a varying number of targets, non-causal, and non-realtime. At first, we choose YOLO V3 to detect the objects included in each frame. Unsuitable candidates were screened out and the rest of detection results are regarded as multiple agents and forming a multi-agent system. Independent Q-Learners (IQL) is used to learn the agents' policy, in which, each agent treats other agents as part of the environment. Then, we conducted offline learning in the training and online learning during the tracking. Our experiments demonstrate that the use of MADRL achieves better performance than the other state-of-art methods in precision, accuracy, and robustness.

INDEX TERMS Multi-object tracking, MADRL, IQL, YOLO V3.

I. INTRODUCTION

Visual multi-object tracking is one of the crucial problems in computer vision field and has a wide range of applications, such as, robotics, artificial intelligence, virtual reality and so on [1]–[4]. Despite great successes in the last decades, multi-object tracking still remains challenging due to a lot of factors including object appearing or disappearing, occlusion, appearance similarity, background clutter [5]–[7].

In recent progress on multi-object tracking, tracking-by-detection strategy has been focused on due to rapid development for object detection methods [8]–[11]. To overcome ambiguities in associating object detections and resolve the detection failures, some research papers take future time steps into account, which are not suitable for online tracking applications, for example, autonomous driving and robot navigation because they are not causal systems [12]–[15].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhihua Qu.

In most of recent works, tracking-by-detection multi-object tracking approaches are roughly divided into two categories: offline mode and online mode [16]–[18]. In offline learning, we can perform learning before the actual tracking happens, in which the detections of all the frames in the video sequence are often used together to avoid detection failures. The offline learning use ground truth of objects' trajectories to complete supervised learning which can prevent tracking drift happening. A cluttered or crowded scene usually brings some difficulties as the offline learning is static and cannot consider the dynamic of the object in the history of data association. To overcome these difficulties, the global data association is used in many multi-object tracking algorithms. However, only using the offline approaches, the tracking performance is still limited and it is hard to be applied to real-time applications.

On the contrary, online methods perform learning during tracking which can be applied to real-time

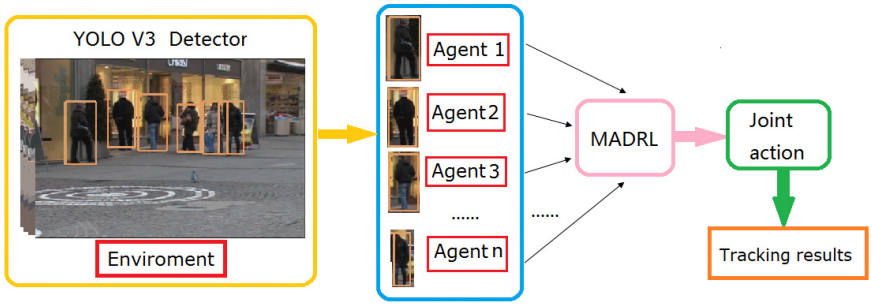


FIGURE 1. Pipeline of multi-object tracking algorithm based on MADRL.

applications [19], [20]. The major challenge is the ambiguities in associating noisy detections in the current frame with the tracked objects in the previous frame. To handle this challenge, different cues, such as motion and appearance, often are combined in association. The hand-crafted features, such as Harr-like features [21], histograms of oriented gradients (HOG) [22], and local binary patterns (LBP) [23] are used frequently in most previous multi-object tracking methods. As more complex characteristics of the objects cannot be captured using hand-crafted features, these exiting methods have many limits in applications. In addition, ground truth is not taken into account for supervision in the online learning, some incorrect training examples may lead to tracking drift.

Reinforcement learning (RL) has gained some success in the previous researches, but these existing approaches have poor scalability and are limited to low-dimensional issues [24]–[26]. There are these limitations mainly because RL has higher complexity. With the rising of deep learning, new solutions have been provided to solve these problems. As deep neural networks can provide powerful function approximation and deep feature representations, deep reinforcement learning (DRL) can perform more effective than RL. Compact low-dimensional features of high-dimensional data (such as images, text, and audio) can be found by deep neural networks automatically, which is the most outstanding contribution of deep learning. In the last few years, DRL has achieved rapid progress and opened entrances to a new perspective on this issue [27]–[29], and has been applied in many emerging domains [30]–[35].

Generally speaking, DRL considers a single agent in a stationary environment, namely single agent deep reinforcement learning (SADRL). By comparison, multi-agent deep reinforcement learning (MADRL) takes multiple agents learning into account and has received an increased amount of attention [36]–[38], but rarely is applied in visual multi-object tracking. Unlike SADRL, converge often fails in MADRL because the objects always move. In MADRL setting, multiple agents' rewards is related to each agent's actions, and finding optimal policies become difficult.

Based on the above analysis, we propose a multi-object tracking algorithm by using MADRL, which can improve the performances in both precision and accuracy. Figure 1 illustrates the pipeline of our proposed tracker, the important contributions can be summarized as follows:

- A tracker based on MADRL is proposed to solve the problems in the existing tracking methods, such as a varying number of targets, non-causal, non-realtime, etc. To the best of our knowledge, we are the first to apply MADRL to solve the problem of visual multi-object tracking.
- In our tracker, YOLO V3 is adopted as object detector as it has state-of-art performance and is a real-time detection system. A single frame image is considered as an environment, each single object is formulated as an agent, a set of agents in the shared environment forms a multi-agent system. IQL is used as it is more practical in processing multi-object tracking problem, in which, each agent learns its own policy independently, and treats other agents as part of the environment.
- Learning a similarity function for data association in multi-object tracking is equivalent to learning a policy in MADRL. We conducted offline learning in the period of training and online learning during the tracking phase, which take full advantage of offline learning and online learning.

The rest of our paper is structured as follows: the background is reviewed in the following section. Section III. introduces our proposed multi-object tracking method. The experimental results and analysis are demonstrated in Section IV. Finally, we draw conclusions in Section V.

II. BACKGROUND

A. SINGLE-AGENT DEEP REINFORCEMENT LEARNING (SADRL)

A traditional RL problem can be described as a Markov decision process (MDP), in which the agent aims to make a sequence decisions. RL provides a coherent framework, an agent can learn from an environment a policy function that maps states to actions and take actions in order to maximize its expected cumulative rewards at each discrete time step.

Formally, RL defines an environment $\boldsymbol{\varepsilon}$, and the state $s \in \mathcal{S}$ of an agent at time step t, the agent need to perform an action $u \in U$, and a reward function R can help the agent to learn

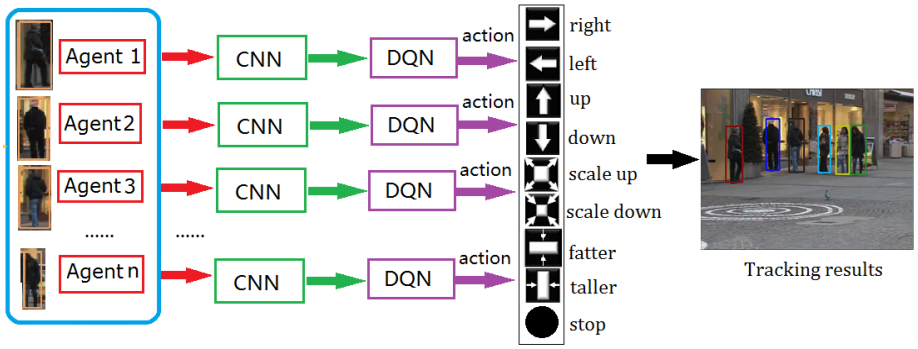


FIGURE 2. Flow chart of the MADRL.

an optimal policy $\pi(a|s)$ to choose an action based on its states. A state transition function $P(s'|s,a)$ which map a pair of state-action at time step t onto a distribution of states at time step $t+1$.

The goal of the agent is to maximize its expected cumulative rewards $R=\sum_{t=0}^{\infty}\gamma^t r_t$, where $\gamma\in[0,1)$ is the discount factor and r_t is a reward signal that the agent receives from the environment at time step t during the training process. In tracking 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 at each time step. When 'stop' action is selected at termination step T, the reward signal r_T is a thresholding function of IoU as follows:

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

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

B. MULTI-AGENT DEEP REINFORCEMENT LEARNING (MADRL)

Different from SADRL, MADRL considers multiple agents learning by RL and the non-stationarity caused by other agents changing their behaviors when they learn. A set of agents in a shared environment, which must learn to maximize their individual returns, are involved in MADRL.

Deep Q-Network (DQN) is one of popular methods that used to find an optimal action-selection policy 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. A recurrent neural network parameterized by θ is usually used to represent the Q-function in deep Q-learning. The action-value function Q of a policy π is:

$$Q^{\pi}(s,a|\theta)=\mathbb{E}[R_t|s_t=s,a_t=a,\pi] \tag{2}$$

Given $Q^{\pi}(s,a|\theta)$, the best policy can be found by

$$Q^{*}(s,a|\theta)=\arg\max_{u} Q^{\pi}(s,a|\theta) \tag{3}$$

The function is defined as the Bellman equation [39] to learn Q^{π} actually, which has the following recursive form,

$$Q^{*}(s,a|\theta)=\mathbb{E}_{S'}[r+\gamma Q^{*}(s',a')|\theta] \tag{4}$$

The agent can choose actions at each time step on the basis of the exploration policy, e.g. an $\varepsilon-$ greedy policy can take the currently estimated best action with probability $1-\varepsilon$, and selects a random exploratory action with probability ε. At each iteration i, experience tuple $\langle s,a,r,s'\rangle$ is stored in a reply memory M and the parameters of DQN θ are updated to minimize the loss function,

$$L(s,a|\theta^i)=\sum_{i=1}^{n}[(r^i+\gamma\max_{a'}Q(s',a'|\hat{\theta}^i)-Q(s,a|\theta^i))^2] \tag{5}$$

The parameters of target network are in combination with experience reply and updated less frequently, that are important for stable deep Q-learning.

III. PROPOSED METHOD

A brief architecture of our proposed multi-object tracking algorithm based on MADRL will be shown in the following subsections firstly. And we will describe the details of our method in the rest of this paper.

A. PIPELINE OF OUR ALGORITHM

The pipeline of our method is demonstrated in Figure 1. Firstly, multiple objects are detected by YOLO V3 [40], which is a state-of-the-art, real-time object detection system. In each frame, YOLO V3 is applied and will output a set of results of detection D_t at time step t, which may include different kinds of objects. We compute the intersecton-over-union (IoU) distance between the ground truth and the results of detection at first frame to get the detections to the tracked. Then, the selected results of object detection are considered as multiple agents and forming a multi-agent system. At last, we adopt a MADRL that can learn to obtain a joint action for multiple objects and get the multi-object tracking results.

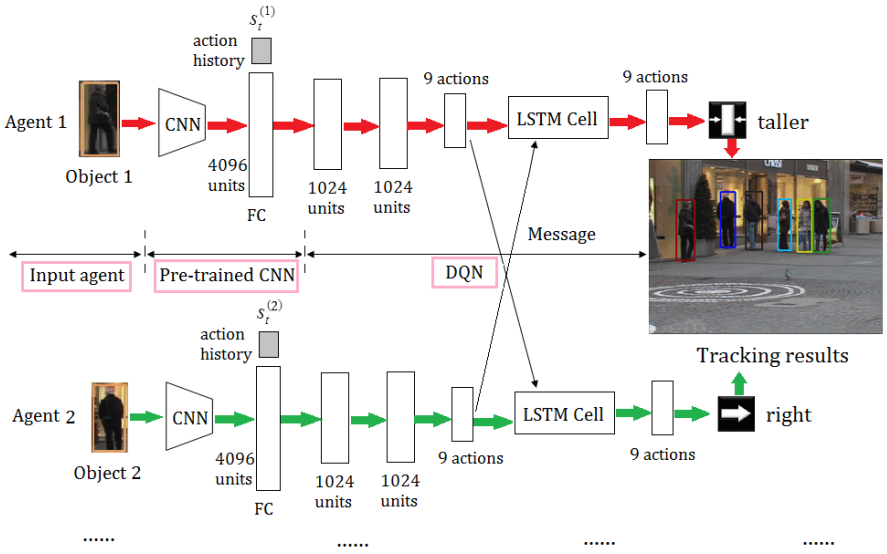


FIGURE 3. The details of the proposed DQN.

B. MULTI-OBJECT TRACKER VIA MADPL

The problem of multi-object tracking is solved by a MADRL method in our tracker. Our MADRL framework is shown as Figure 2, and details of these components will be presented in this section.

In our formulation, we consider a single frame image as an environment. In a multi-agent setting n agents are described by $i \in I \equiv \{1, \ldots, n\}$, each agent takes a set of actions,forming a joint action $a \in A \equiv A^n$, to achieve its goal, the agent 's information of the current environment is represented by a set of states $s \in \mathcal{S}$, state transition probabilities are defined by $P(s' \mid s, a)$. Each agent's observations $z \in Z$ are governed by an observation function $O(s, a)$. The ith agent i selects its action based on its own action-observation history $\tau_i \in T$ according to its policy $\pi(a^i \mid \tau^i)$. After each state transiton, the new observation $O(s, u)$ and the action a^i are added to τ^i, then $\tau^{i\prime}$ come into being.

In our method, we adopt the deep Q-learning algorithm, and the details of the proposed DQN are illustrated as Figure 3. The input agent is processed by a pre-trained CNN, which is conducted on the VGG-16 network and includes five pooling stages and one fully connected layer, i.e. Conv1-2, Conv2-2, Conv3-3, Conv4-3, Conv5-3. The output of the CNN is the state representation of the agent, which is concatenated with the action history. Then, it is input into the DQN which can output the prediction of the value of the actions. The value of actions are applied to the bounding box which is composed of eight actions and one action to terminate the tracking process. Each action is encoded by the 9-dimensional vector, which are defined as follows: move right, move left, move up, move down, scale up, scale down, aspect ratio change fatter, aspect ratio change taller, stop. Motivated by [41], we adopt LSTM cells to exchange messages among the agents.

Suppose the Q-network function of the ith agent is $Q(s^i, a^i \mid \theta_a^i)$, the inter-agent communication is $Q(s^i, a^i, m^i, m^{-i} \mid \theta_a^i, \theta_m^i)$, where m^i represents the messages that sent out from agent i, and m^{-i} the messages that agent i received from other agents. The message is formalized as a function $m(s, a \mid \theta_m)$, where θ_m is learned by using deep learning method, which is outperform handcrafted features.

C. MULTI-AGENT IMPORTANCE SAMPLING AND LEARNING

MADRL falls into two major categories: Independent Q-Learners(IQL) and Joint Action Learners(JAL). Only local actions are observed by the agent in IQL, and actions taken by all agents are observed in JAL. IQL is utilized in our approach as it is more practical in processing multi-object tracking problem. In IQL, each agent learns its own policy independently, and treats other agents as part of the environment. However, IQL introduces an important problem: the environment becomes non-stationary from agents' local perspectives due to multiple agents' the interactions with the environment. Each agent has to coordinate with fellow agents so that MADRL has higher effectiveness.

In our method, the non-stationary that caused by IQL is addressed by adopting an importance sampling scheme for the multi-agent setting. In MADRL, we sample the action a_t^i of agent i at time step t, and sample all the agents, according to both the messages sent out from itself and from other agents. According to Eq.5, we can find that the goal of updating the parameters of the Q-network is to minimize the following importance loss function for agent i :

$$L(s_t^i, a_t^i \mid \theta_a^i, \theta_m^i) = \sum_{i=1}^{n} [(r_t^i + \gamma \max_{a'^i} Q(s_{t+1}^i, a'^i \mid \widehat{\theta}_a^i, \widehat{\theta}_m^i) - Q(s_t^i, a_t^i, m_t^i, m_t^{-i} \mid \theta_a^i, \theta_m^i))^2] \quad (6)$$

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

Test videos	Trackers	MOTA%	MOTP%
Venice-1	MADRL(Our tracker)	26.5	73.7
	RNN-LSTM	12.7	71.7
	SiameseCNN	22.3	73.0
	MDPSubCNN	15.9	72.4
	LP_SSVM	17.8	73.0
ADL-Rundle-3	MADRL(Our tracker)	46.7	79.4
	RNN-LSTM	23.7	72.0
	SiameseCNN	39.7	72.9
	MDPSubCNN	44.9	79.6
	LP_SSVM	28.0	72.9
AVG-Town Center	MADRL(Our tracker)	49.8	73.5
	RNN-LSTM	13.4	68.8
	SiameseCNN	19.3	69.0
	MDPSubCNN	49.5	70.1
	LP_SSVM	14.7	70.1
AVG-Town Center	MADRL(Our tracker)	30.5	76.4
	RNN-LSTM	21.1	75.5
	SiameseCNN	27.5	74.1
	MDPSubCNN	28.8	74.7
	LP_SSVM	24.9	75.6
ETH-Linthescher	MADRL(Our tracker)	28.3	75.1
	RNN-LSTM	12.4	74.7
	SiameseCNN	16.7	74.2
	MDPSubCNN	27.2	74.7
	LP_SSVM	15.6	75.6
ETH-Jelmoli	MADRL(Our tracker)	41.9	75.1
	RNN-LSTM	34.8	73.3
	SiameseCNN	42.3	72.8
	MDPSubCNN	32.9	73.6
	LP_SSVM	39.5	74.4
PETS09-S2L2	MADRL(Our tracker)	47.9	73.6
	RNN-LSTM	38.3	71.6
	SiameseCNN	47.5	72.6
	MDPSubCNN	34.5	69.7
	LP_SSVM	41.5	70.5
TUD-Crossing	MADRL(Our tracker)	79.6	77.3
	RNN-LSTM	57.2	71.7
	SiameseCNN	73.7	73.0
	MDPSubCNN	78.9	76.7
	LP_SSVM	60.0	74.2

Learning a similarity function for data association in multi-object tracking is equivalent to learning a policy in MADRL. Motivated by [42], we conducted offline learning in the period of training and online-learning during the tracking phase, because the ground truth can be used for supervision to avoid the tracking drift in offline learning, at the same time, the dynamic status and the history of the target object can be taken into account in online-learning.

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

The experiments of our proposed multi-object tracking algorithm were conducted on a workstation equipped with the Windows 10 operating system, Intel(R) Core(TM) i7-4712MQ CPU, 32GB RAM, and GeForce GTX TITAN X GPU, 12.00 GB VRAM. We used MATLAB R2016b as our software platform. In CNN, the learning rate is set to 0.0001 for convolutional layers and is set to 0.001 for fully-connected layers.

B. QUANTITATIVE EVALUATION

In this section, our approach (MADRL) is compared with other five state-of-art multi-object trackers, i.e. MDPSubCNN [42], RNN-LSTM [43], SiameseCNN [44], LP_SSVM [45], LSTM_DRL [46], on the MOT challenge benchmark [47] in order to evaluate the tracking performance. We use the CLEAR MOT metrics for quantitative evaluation including the multiple object tracking accuracy (MOTA), the multiple object tracking precision (MOTP). On the 8 test videos that have public CLEAR MOT metrics data included in the MOT Challenge dataset, the

FIGURE 4. Sample tracking results on the MOT challenge benchmark.

quantitative comparison is conducted between our tracker with other state-of-art trackers, and the results are reported in Table 1.

TABLE 2. The evaluation results of running time.

Trakers	Running time
MADRL(Our tracker)	110.4fps
RNN-LSTM	166.8fps
MDPSubCNN	2.1fps

C. QUALITATIVE EVALUATION

Due to the limited given space, we only list the part of tracking results on the test videos in the MOT challenge benchmark, as demonstrated in Figure 4.

D. EVALUATION OF RUNNING TIME

We conducted running time evaluation using the above workstation equipped with GeForce GTX TITAN X GPU. We compare our tracker to other two state of-the-art trackers and list the results of running time on the MOT Challenge benchmark in Table 2. Our method is not the fastest tracker but a real-time tracking system. RNN-LSTM obtains the fastest speed as it does not incorporate appearance, and our tracker is better than it on the other performances.

The above experimental data listed in the Table1–2 and Figure 4 demonstrate the superior performance of our track strategy with MADRL in both precision and success rate.

V. CONCLUSION

There are some problems in the existing multi-object trackers, for example, they fails when the object emerging or disappearing, there are many limitations as complex characteristics of the objects can not be captured by the hand-crafted features, the tracked objects have similar appearance, etc.. To overcome these problems, a novel multi-object tracking approach based on MADRL was proposed in this paper. The object detector YOLO V3 was adopted to detect the multiple objects. The detected results is considered as multiple agents, then, we adopt a MADRL to obtain a joint action for multiple objects and get the multi-object tracking results. The experimental results showed that the proposed multi-object tracking method obtains the better performances in the robustness and accuracy.

REFERENCES

[1] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3786–3795.

[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, pp. 121–138, Jan. 2017.

[3] A. Kumar K.C., L. Jacques, and C. De Vleeschouwer, "Discriminative and efficient label propagation on complementary graphs for multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 61–74, Jan. 2017.

[4] J. R. B. Del Rosario, A. A. Bandala, and E. P. Dadios, "Multi-view multi-object tracking in an intelligent transportation system: A literature review," in *Proc. IEEE 9th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Dec. 2017, pp. 1–4.

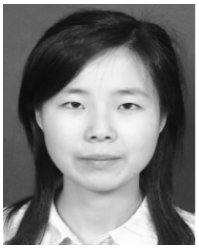
[5] M. A. Naiel, M. O. Ahmad, M. N. S. Swamy, J. Lim, and M.-H. Yang, "Online multi-object tracking via robust collaborative model and sample selection," *Comput. Vis. Image Understand.*, vol. 154, pp. 94–107, Jan. 2017.
[6] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1614–1627, Aug. 2014.
[7] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6951–6960.
[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[9] A. Andriyenko, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2054–2066, Oct. 2016.
[10] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[12] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.
[13] J. F. Henriques, R. Caseiro, and J. Batista, "Globally optimal solution to multi-object tracking with merged measurements," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2470–2477.
[14] A. A. Butt and R. T. Collins, "Multi-target tracking by lagrangian relaxation to min-cost network flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1846–1853.
[15] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1948–1955.
[16] S. Bae and K. Yoon "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 595–610, Mar. 2018
[17] L. Wen, Z. Lei, S. Lyu, S. Z. Li, and M. Yang, "Exploiting hierarchical dense structures on hypergraphs for multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1983–1996, Oct. 2016.
[18] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011.
[19] Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang, "Connected component model for multi-object tracking," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3698–3711, Aug. 2016.
[20] A. Dehghan, Y. Tian, P. H. S. Torr, and M. Shah, "Target identity-aware network flow for online multiple target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1146–1154.
[21] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
[22] N. Dalal and B. Triggs "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.
[23] X. Wang, T. X. Han, and S. Yan "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV)*, Kyoto, Japan, Sep./Oct. 2009, pp. 32–39.
[24] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.
[25] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[26] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr./May 2004, pp. 2619–2624.
[27] Y. Li. (2017). "Deep reinforcement learning: An overview." [Online]. Available: https://arxiv.org/abs/1701.07274
[28] D. Jayaraman and K. Grauman. (2016). "Look-ahead before you leap: End-to-end active recognition by forecasting the effect of motion." [Online]. Available: https://arxiv.org/abs/1605.00164
[29] D. Silver, *et al.* "Mastering the game of go with deep neural networks and tree search," *Nature* vol. 529, no. 7587, pp. 484–489, Jan. 2016.
[30] D. Zhang, H. Maei, X. Wang, and Y.-F. Wang. (2017). "Deep reinforcement learning for visual object tracking in videos." [Online]. Available: https://arxiv.org/abs/1701.08936
[31] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang. (2017). "End-to-end active object tracking via reinforcement learning." [Online]. Available: https://arxiv.org/abs/1705.10561
[32] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1349–1358.
[33] D. Jayaraman and K. Grauman. (2016). "Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion." [Online]. Available: https://arxiv.org/abs/1605.00164
[34] Z. Jie, X. Liang, J. Feng, X. Jin, W. F. Lu, and S. Yan, "Tree-structured reinforcement learning for sequential object localization," in *Proc. Adv. Neural Inf. Process.*, 2016, pp. 127–135.
[35] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proc. CVPR*, Dec. 2015, pp. 2488–2496.
[36] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, "Evolutionary dynamics of multi-agent learning: A survey," *J. Artif. Intell. Res.*, vol. 53, pp. 659–697, Aug. 2015.
[37] L. Buçoniu, R. BabuŠka, and B. De Schutter, *Multi-agent Reinforcement Learning: An Overview* (Studies in Computational Intelligence), vol. 310. 2010, pp. 183–221. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-642-14435-6_7
[38] D. B. Noureddine, A. Gharbi, and S. B. Ahmed, "Multi-agent deep reinforcement learning for task allocation in dynamic environment," in *Proc. Int. Conf. Softw. Technol.*, Jan. 2017, pp. 17–26.
[39] R. Bellman, "On the theory of dynamic programming," *Proc. Nat. Acad. Sci. USA*, vol. 38, no. 8, pp. 716–719, Aug. 1952.
[40] J. Redmon and A. Farhadi. (2018). *YOLOv3: An Incremental Improvement*. [Online]. Available: https://arxiv.org/abs/1804.02767
[41] X. Kong, B. Xin, Y. Wang, and G. Hua, "Collaborative deep reinforcement learning for joint object search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1695–1704.
[42] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4705–4713.
[43] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networksm," in *Proc. AAAI*, Feb. 2017, pp. 4225–4232.
[44] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler "Learning by tracking: Siamese CNN for robust target association. in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 33–40.
[45] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *Int. J. Comput. Vis.*, vol. 122, no. 3, pp. 484–501, May 2017.
[46] M. X. Z. G. X. L. F. Jiang Deng Ch Pan Chen Wang and X. Sun, "Multiple object tracking in videos based on LSTM and deep reinforcement learning," *Complexity*, vol. 2018, Nov. 2018, Art. no. 4695890. [Online]. Available: https://www.hindawi.com/journals/complexity/2018/4695890/
[47] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. (2015). "MOTChallenge 2015: Towards a benchmark for multi-target tracking," [Online]. Available: https://arxiv.org/abs/1504.01942
[48] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3047–3055.

MINGXIN JIANG received the B.S. degree in measurement and control technology and instrument and the M.S. degree in communications and information system from Jilin University, Changchun, China, in 2002 and 2005, respectively, and the Ph.D. degree in signal and information processing from the Dalian University of Technology, China, in 2013. She was a Postdoctoral Researcher with the Department of Electrical Engineering, Dalian University of Technology, from 2013 to 2015. She is currently an Associate Professor with the Faculty of Electronic Information Engineering, Huaiyin Institute of Technology. Her research interests include multi-object tracking, video content analysis, and vision sensors for robotics.

TAO HAI received the B.Sc. degree from the Department of Computer and Information Science, Northwest University of Nationalities, in 2004, the M.S. degree from the School of Mathematics and Statistics, Lanzhou University, in 2009, and the Ph.D. degree from the Faculty of Computer System and Software Engineering, Universiti Malaysia Pahang, in 2012. He is currently an Associate Professor with the Baoji University of Arts and Sciences. His current research interests include machine learning, the Internet of Things, and optimization computation.

ZHIGENG PAN received the Ph.D. degree in computer graphics from Zhejiang University. He is currently the Director of the Digital Media and HCI Research Center, Hangzhou Normal University. His research interests include virtual reality, computer graphics, and human-computer interaction. He is a member of ACM SIGGRAPH.

HAIYAN WANG received the B.S. degree in electronic information science and technology from Huaiyin Normal University, Huaian, China, in 2004, and the M.E. degree in communications and signal processing from the Guilin University of Electronic Technology, Guilin, China, in 2012. She is currently with the Faculty of Electronic Information Engineering, Huaiyin Institute of Technology, Huaian. Her research interests include computer vision and signal processing.



YINJIE JIA is currently pursuing the Ph.D. degree with the College of Computer and Information Engineering, Hohai University, Nanjing, China. Since 2003, she has been teaching in the Huaiyin Institute of Technology. Her current research interests include image processing, multimedia information processing, and computer vision.

CHAO DENG received the B.S. degree and the M.S. degree in communication engineering from Jilin University, China, in 2002 and 2005, respectively, and the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2008. He is currently an Associate Professor with the School of Physics and Electronic Information Engineering, Henan Polytechnic University, Jiaozuo, China. His research interests include image processing and signal processing.