IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 22, 2022, accepted June 7, 2022, date of publication June 13, 2022, date of current version June 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3182792

Object Tracking Using Siamese Network-Based Reinforcement Learning

SUNG JUN PARK, SEUNG JUN HWANG, AND JOONG-HWAN BAEK
School of Electronics and Information Engineering, Korea Aerospace University, Goyang 10540, South Korea

Corresponding author: Joong-Hwan Baek (jhbaek@kau.ac.kr)

This work was supported by the Gyeonggi-do Regional Research Center (GRRC) Program of Gyeonggi Province, Development of Intelligent Interactive Media and Space Convergence Application System, under Grant GRRC Aviation 2017-B04.

ABSTRACT Object tracking is a technique for tracking a specific object appearing in a video sequence while observing its features or changes. Recently, many algorithms showing high performance have emerged by applying the Siamese network to the object tracking field. A Siamese network is designed to learn the similarity between two images. In object tracking, the Siamese network tracks the object by finding the location most similar to the target image in the search image. Algorithms based on Siamese networks are vulnerable to partial and total occlusion of objects. In addition, since the object is tracked using only the similarity with the image obtained using the ground-truth bounding box of the first frame, if an object is missed once, then errors are accumulated, and a situation where the object drifts away from the object of interest frequently occurs. Therefore, in this paper, we propose a reinforcement learning model that can maximize the reward for tracking success after partial and total occlusion of an object. We also propose a dynamic template exchange method using a template that has been successfully tracked in a recent frame to solve the drift problem. When the proposed model is applied to the existing tracking models to evaluate the quantitative performance in representative object tracking benchmarks VOT2018 and OTB50, it is confirmed that the accuracy is improved, and the number of tracking failures decreases compared to the existing method. As a result, an accuracy of 0.618, robustness of 0.234, and expected average overlap (EAO) of 0.416 are achieved in VOT2018, and success of 0.673 and precision of 0.881 are achieved in OTB50.

INDEX TERMS Object tracking, Siamese network, region proposal network, reinforcement learning, dynamic template exchange.

I. INTRODUCTION

Visual object tracking is a fundamental computer vision task. In this field, it is possible to infer the correlation of target objects between frames in a video sequence. It is used as a basic work of video application in fields such as robot vision [1], [2], self-driving [3], [4], and surveillance systems [5], [6]. Although tracking algorithms are used in various applications, problems such as partial and full occlusion of objects, scale changes, and object/camera motion remain challenges to be solved [7]. That is, spatial features and temporal features must be present so that the initially selected object of interest can be tracked to the end. It is necessary to solve the occlusion and drift problems that occur during the tracking process. However, occlusion is difficult to define for annotating as ground-truth in the training dataset. Most of the training datasets constructed thus far are only annotated with 1 or 0 in the frame in which the occlusion occurs. We need data or a model that can effectively learn about the occlusion and drift. Reinforcement learning is mainly used in tasks where training data are scarce or ground-truth setting is difficult. In object tracking, reinforcement learning can experience success and failure through tracking simulation. Therefore, in this paper, we propose a reinforcement learning model that can be applied to the existing tracker by defining the state, action, and reward to solve the occlusion and drift problems. Our proposed reinforcement learning model integrates the channels divided into foreground and background into a single channel. Then, the agent learns to select where the tracking can be successful in the feature of the occlusion situation. Existing methods using reinforcement learning to be described in Section 2 are designed to move

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

the bounding box. The model then has a prior experience of the location of the bounding box. This tends to keep track of the intact objects, and it is more likely to lose the target object in the event of an occlusion. The proposed method in this paper allows the agent to pre-experience tracking hindrances such as occlusion and drift. From the feature map at that time, a feature that can be successfully tracked is selected. This can be learned in a way that maximizes the rewards the agent can earn in a reinforcement learning environment.

Recently, the Siamese network [8] has been applied to tracking tasks, showing balanced performance in speed and accuracy, and various applications are continuing. Typically, in [9]–[12], the ground-truth of the object of interest in the first frame was maintained as a template, and the object was tracked until the end of the video sequence. These models were designed as CNNs, so the model was mainly used to capture spatial features.

It is difficult to solve the continuous tracking problem caused by temporal features within a sequence [13]. References [14]–[16] solved it by matching several templates with the object of interest during tracking. However, a model for this needs to be additionally designed. To simplify this, in this paper, the dynamic template exchange method of Yan *et al.* [13] is applied to a Siamese network to enable the capture of temporal characteristic information, thereby solving the temporal problem.

In general, object tracking models are trained using the coordinates of the bounding box representing the object location. The predicted coordinates have various influences, such as the starting point for inference in the next frame and the motion model. The tracking model makes inferences every frame. This is why a single inference affects the tracking until the end of the sequence. When an occlusion occurs, the tracking of a part of the object causes errors to accumulate and drift or leads to tracking success. Even if the overlap ratio between the ground-truth and the estimated result in one frame is measured to be high, it may not be a successful inference in the entire sequence. Fig. 1 shows the tracking results of the red box tracker and the green box tracker initialized with the ground-truth (cyan box) in the first column. The green box tracking results (GOTURN [17], ATOM [18], and DiMP50 [19]) in the second column have a higher intersection over union (IoU) with the ground-truth than the red-box tracking result. In the second column, the IoU value between the red box and ground-truth falls below 0.5. This means that tracking fails. However, similar to the third and fourth columns, when the occlusion of the tracking object is finished, the complete object shape is found due to the position of the bounding box, and when the sequence ends, the tracking can be successful. In this way, it can be confirmed that the inference result for every frame has a great influence on the accuracy and robustness of the tracking model. As with most tracking models, pretrained models and CNNs in the backbone network tend to track larger and intact objects. When the tracking object is obscured by other objects, it will track other objects that appear intact. Therefore, tracking results such as green boxes occur frequently.

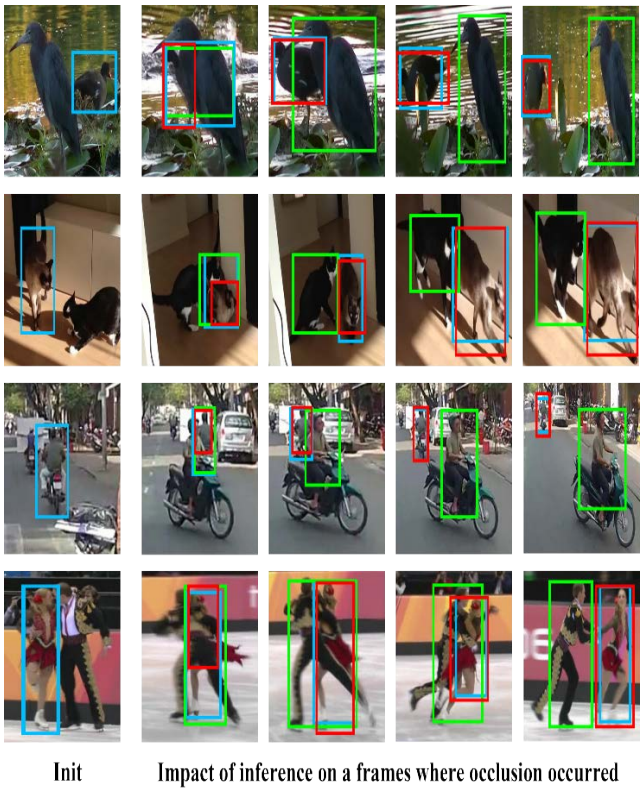

FIGURE 1. Examples of inference in frame where occlusion occurred in last frame. Cyan box is ground-truth, green box is CNN-based tracker results (GOTURN – 1st and 3rd rows, ATOM – 2nd row, DiMP50 – 4th row), and red box is ours. In second column where occlusion occurred, green box performed better than red box based on IoU. However, in second column, red box chose location for successful tracking in subsequent frames. As a result, red box shows continuous success in tracking, but green box fails.

In this paper, to solve this problem, the tracking performance is improved by learning the experience of tracking success and failure through a reinforcement learning model that rewards when tracking is successful in the last frame of the sequence and gives a penalty when it fails. By combining a Siamese network and a region proposal network, the similarity score map output from the object tracking model and the vector for the moving direction of the object are set as the state, and the selection of the location of the object on the score map is set as the action. The reward is given according to the success of tracking the last frame in the learning sequence. As a result, state-of-the-art performance is achieved by linking the reinforcement learning model and dynamic template exchange method proposed in the VOT2018 [20] and OTB50 [21] benchmarks with the existing tracker.

The main contributions of this work are as follows:

- We propose a new reinforcement learning framework to solve the occlusion problem.

- We propose a dynamic template exchange method applicable to Siamese network-based tracking algorithms to solve the drift problem.
- Our proposed method outperforms the state-of-the-art methods in VOT2018 and OTB50.

The structure of this paper is as follows. Section 2 introduces the study of applying visual object tracking and deep reinforcement learning to the tracking model. Section 3 introduces the reinforcement learning model proposed in this study, the problem settings in reinforcement learning, and the dynamic template exchange method in the Siamese network. Section 4 compares the performance with existing studies on the tracking benchmark and verifies that the performance is improved. Finally, a conclusion is drawn in Section 5.

II. RELATED WORKS

The purpose of this study is to improve the performance of the existing Siamese network-based object tracking model using reinforcement learning. In this section, we review the existing object tracking model and representative methods in the field of object tracking using reinforcement learning. In the object tracking section, we briefly introduce CNN-based tracking models. Additionally, the object tracking models used as the starting point of this paper, SiamRPN [10], SiamRPN++ [11], and SiamMask [12], are explained. In the reinforcement learning-based object tracking session, existing studies are described on how the reinforcement learning model is applied to the object tracking model.

A. VISUAL TRACKING

Until the Siamese network-based object tracking model was developed, many tracking models using the basic structure of CNN were developed. C-COT [22] proposed a new structure that uses a continuous convolution filter instead of a discriminative correlation filter for learning, greatly improving the tracking performance. ECO [23] improved the accuracy and speed performance by optimizing the key factors that degrade the tracking performance in C-COT. MDNet [24] significantly improved tracking performance by suggesting shared layers to obtain a general target representation and a domain-specific layer structure for a binary classifier that identifies targets in each domain. VITAL [25] proposed a new structure for applying a generative adversarial network [26] to object tracking. GOTURN [17] proposed a structure that is similar to the Siamese network but shares features extracted from CNNs from two input images in a fully connected layer. Although the above CNN-based tracking algorithms have been proposed in various network structures, SiamFC [9], a tracking model based on the Siamese network, shows a balanced performance in terms of accuracy and speed and changes the paradigm of the object tracking algorithm. SiamFC is a study aimed at proving the efficiency of the Siamese network. Without adding any additional cues, the output of the model was used without bounding box regression. Therefore, various studies (SiamVGG [27], SiamFC [28], and SiamDW [29]) were conducted based on SiamFC research, and its application to thermal infrared images (HSSNet [30], MLSSNet [31], and MMNet [32]) as well as RGB images shows high performance. Among them, SiamRPN [10], which applied the region proposal network [33] to SiamFC, and SiamMask, which added a mask branch to SiamRPN, are representative algorithms that greatly improved the performance of the Siamese network-based object tracking algorithm. The Siamese network framework as above has been used as a starting point for various studies (SiamMask_E [34], THOR [16], and Siam R-CNN [35]) until recently. In this paper, SiamMask, which shows higher performance, is used as the starting point.

First, SiamRPN [10] inputs the results of the Siamese network to the classification and regression branches of the regional proposal network. Then it outputs k anchor box positions and classification scores for objects and backgrounds through cross-correlation. Because only offline learning is performed, it shows a fairly high-speed performance. During training, the classification branch is output as two channels (positive and negative) for each anchor, and cross-entropy loss (L_{cls}) is used. In the regression branch, the center coordinates, width, and height of each anchor are output to 4 channels (dx, dy, dw, and dh). The loss function is used by adopting the $smoothL_1$ loss function (L_{reg}). The input to the loss function is the normalized coordinates (δ) of the ground-truth box (G) and the anchor box (AN) defined as in (1). Finally, the total loss L_{SRPN} of SiamRPN is the same as (2). Here, $\lambda(\geq 0)$ is a hyperparameter.

$$\begin{aligned} \delta[0] &= \frac{G_x - AN_x}{A_w} \delta[1] = \frac{G_y - AN_y}{A_h} \\ \delta[2] &= \ln \frac{G_w}{AN_w} \delta[3] = \ln \frac{G_h}{AN_h} \end{aligned} \tag{1}$$

$$L_{SRPN} = L_{cls} + \lambda L_{reg} \tag{2}$$

SiamRPN++ [11] is an improved model that explores some disadvantages of SiamRPN. It shows the highest performance among contemporary object tracking models by removing padding to maintain spatial invariance and reducing parameters by changing the cross-correlation method to depthwise cross-correlation.

SiamMask extends the mask branch and loss function to SiamRPN to encode the features required for outputting the binary segmentation mask of an object. Since it is possible to obtain a mask for an object, the gap between Visual Object Tracking and Video Object Segmentation is reduced, and tracking performance is greatly improved. In the binary mask, a target image (z) and a search image (x) are output through a mask branch (M_ϕ). That is, the binary mask corresponding to the feature map obtained by the depthwise cross-correlation layer can be expressed as shown in (3) so that it is possible to generate another mask for the search image.

$$Mask = M_\phi(z, x) \tag{3}$$

The loss function for the mask during training is defined in the form of a logistic loss function between the pixel-by-pixel

annotated ground-truth and the predicted mask. Finally, SiamMask's total loss L_{SM} adds a loss function (L_{mask}) for the mask branch to the two loss functions used for SiamRPN training. As shown in (4), the model is trained using COCO [36], ImageNet-VID [37], and YouTube-VOS [38] in combination with the hyperparameters of Pinheiro *et al.* [39]. Here, $\lambda_1 = 32$ and $\lambda_2 = \lambda_3 = 1$ are set.

$$L_{SM} = \lambda_1 L_{mask} + \lambda_2 L_{cls} + \lambda_3 L_{reg} \tag{4}$$

In inference, the binary mask of the object is predicted at the index that outputs the highest score in the score branch. The search region is cropped by referring to the bounding box location from the box branch of the corresponding index. Although it showed high performance in the VOT2018 benchmark, tracking performance is still poor when tracking motion blur and nonobjects. The reason is that, as the author mentioned, the training dataset focused on intact objects. Supervised learning repeatedly learns well-refined data despite the use of data augmentation. Because it learns while reducing the error between the inferred result and ground-truth on the annotated data, it self-limits the actual test data. As mentioned in Section 1, this was overcome by performing tracking without ground-truth through tracking simulation in reinforcement learning. Yan *et al.* [13] presented the problem that if only convolutional operation is used, training on temporal features is difficult and vulnerable to large-scale changes in objects. To solve this problem, Transformer [40], which is mainly used in the natural language processing (NLP) task, is used. In this paper, the concept of a dynamic template proposed by Yan *et al.* is applied to a Siamese convolutional network to overcome the problem of capturing temporal features.

B. REINFORCEMENT LEARNING

ADNet [38], the most representative algorithm applying reinforcement learning to object tracking, was a great inspiration for this study. Yun *et al.* [41] pointed out the inefficiency of the search algorithm of MDNet. This is because MDNet selects the best candidate by matching the tracking model after searching the region of interest. In addition, the problem that labels should be annotated on all frames to train the model was presented. To solve this problem, an algorithm combining supervised learning and reinforcement learning was proposed. Silver *et al.* [42] showed a study result that the performance of the reinforcement learning policy network can be significantly improved if it is pretrained through supervised learning. Similarly, in ADNet, the parameters of the network were updated through reinforcement learning after supervised learning by annotating labels on actions according to states. ADNet tracks the object by controlling the bounding box being expressed in the sequence through successive actions selected by the model. Action is an 11-dimensional vector, and the movement and scale adjustment of the bounding box are defined as shown in Fig. 2. A state is defined as a tuple of vectors containing the image patch within the bounding box and the previous 10 actions. When the model chooses

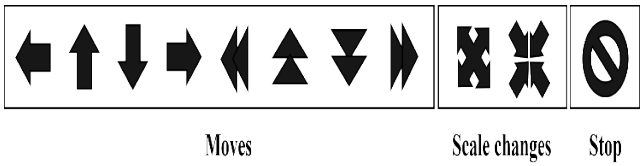

FIGURE 2. ADNet's action set.

a stop action, the agent is rewarded and then transitioned to the initial state in the next frame. The parameters of the model are updated through stochastic gradient ascent (SGA) [43] to give rewards by comparing the results of sequential actions and IoU with ground-truth and maximizing the rewards. Additionally, even if the ground-truth exists intermittently in the video sequence, we need to reward only the frame where the ground-truth exists. Due to this advantage, semi-supervised learning can overcome the limitations of test data.

Zhang *et al.* [44] proposed a method to learn spatial and temporal information by applying reinforcement learning to a network combined with CNN and LSTM [45]. Similar to ADNet, they used the reinforcement learning algorithm proposed by Williams [43] and designed a CNN to encode the features extracted from the frame and an RNN to regress the position of the target object in time steps.

TRASFUST [46] designed a model by combining knowledge distillation (KD) [47] and reinforcement learning. TRASFUST defines a state as a patch of two images in a bounding box. Different from ADNet for action, the amount of change for the motion of the bounding box was set as a vector. Using KCF [48], MDNet [24], ECO [23], and SiamRPN [10], which have significantly improved performance in the tracking field, as a Teacher network, the teacher learns the movement of the bounding box predicted by the teacher, and the student transitions the state. This showed state-of-the-art performance against benchmarks such as GOT-10k [49] and UAV123 [50] but showed low performance in VOT2019 [51]. This is because the VOT2019 benchmark was built to evaluate which algorithm estimates the best bounding box by defining the center of an object rather than an intact object as a ground-truth bounding box. However, since TRASFUST tends to track whole objects, it has the same effect as having better performance than other object tracking algorithms in qualitative evaluation.

As mentioned above, object tracking algorithms using reinforcement learning have been actively studied. Most object tracking models using reinforcement learning are designed to refine the location of bounding boxes. However, in this paper, it is designed to select a better feature rather than refine the position of the box. The complexity of the input image is reduced by using the feature map of the score branch output by the Siamese network-based tracking model as input. Similar to ADNet, the performance of the tracking model was improved by designing to maximize the reward for tracking success.

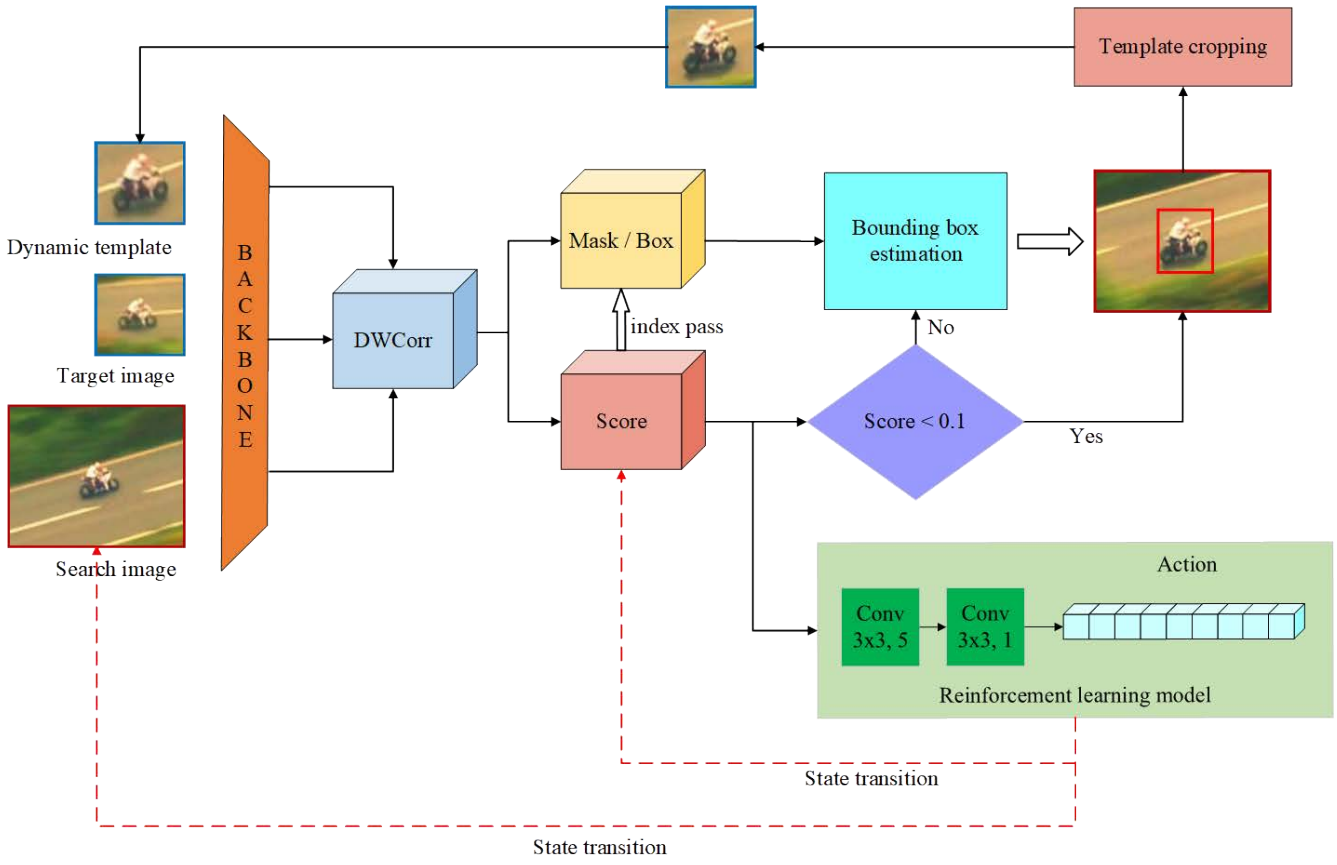

FIGURE 3. Proposed model architecture. The similarity score computed by the tracker's depthwise cross-correlation layer is used as input to the reinforcement learning model and the dynamic template method. The output of the reinforcement learning model becomes the index of the score map and is passed to the mask/box branch for the final bounding box. In the next frame, the result of cropping around the output bounding box is used as the search image. When the value of the selected score is less than 0.1, the tracker determines that the tracking object has been lost and uses the bounding box previously tracked with the highest score as a template. After that, when the template is input to the tracker, a higher similarity score is used as an input for reinforcement learning by comparing the similarity between dynamic template and search image, and between target image and search image.

III. PROPOSED METHODS

In this section, we first describe the model structure of the reinforcement learning framework proposed in Section A. It takes the score map of the Siamese-network-based tracker as input, passes through two convolution layers, and outputs the action to succeed in tracking until the end of the sequence. Section B describes how to define the problems of state, action, and reward in reinforcement learning. Finally, the detailed implementation, learning method, and reasoning process are described.

A. PROPOSED MODEL

The Siamese network-based tracking algorithm tracks the object only with similarity to the ground-truth of the first frame. As a result, if the model misses a tracking object once, then errors accumulate and tend to drift in the wrong place. To compensate for these shortcomings, this paper proposes a tracking model by applying reinforcement learning to the Siamese network-based RPN's score branch. In addition, by applying a dynamic template exchange to the Siamese network-based tracking model, it is designed to ensure accurate tracking by increasing the robustness to temporal variation of objects. Fig. 3 shows the proposed model structure. The reinforcement learning model in this paper follows the Markov decision process (MDP) strategy. The state of the MDP is defined by $s \in S$, the action as $a \in A$, the state transition function as f, and the reward as R. Reinforcement learning performs well in games with similar backgrounds, similar objects, and set rules. However, in object tracking, numerous objects and backgrounds appear in the input image. Therefore, if the input image is used for reinforcement learning as it is, then an infinite state is created, so it is difficult to determine an action according to the state. To solve this problem, the input of the reinforcement learning model should be simplified as much as possible. Therefore, the features extracted from the score map are used as input to the reinforcement learning model. Fig. 4 is an example of a score map used as an input.

The state is set by the score map and the movement direction of the object. Here, the movement direction of the object is used to weight the score map. Furthermore, as shown in

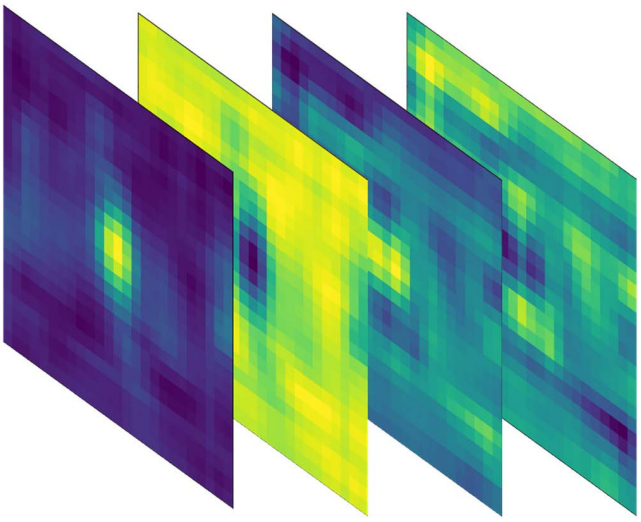

FIGURE 4. Sample of score map.

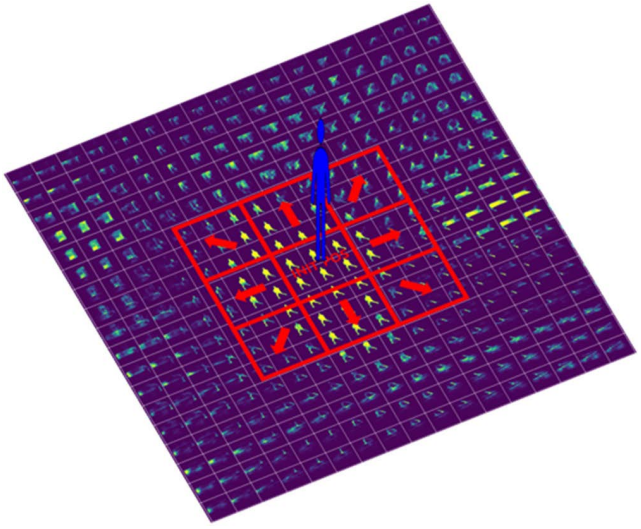

FIGURE 5. A game abstraction in which players choose actions to achieve high scores within a score map.

Fig. 5, the action is defined to select whether the character of the game will move in one of eight directions from the initial position. If the tracking is successful in the last frame of the sequence, then it is designed to abstract the object tracking process like a simple game by giving rewards as if we obtain a score when we clear the stage of the game. The final bounding box is output by passing the index to the box or mask branch according to the selected action. The background in Fig. 5 is a mask according to the selected index. Here, it can be seen that when a low score index is selected, a mask containing a background is obtained rather than a human-shaped mask that was a tracking object.

In Section B, the action, state, state transition function, and reward are described in detail.

B. PROBLEM SETTINGS

1) ACTION

Action is defined in 9 discrete spaces as in (6). (5) is a position where the highest score is output in the score branch when the target image z and the search image x are input. Action A is defined as a 9-dimensional vector with the position of (5) and 8 adjacent positions of the same channel as in (6).

$$m, n = argmax(F_{score}(z, x)) \tag{5}$$

$$A = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_7 \\ a_8 \end{bmatrix} \tag{6}$$

where $a_0 = (m-1, n-1), a_1 = (m-1, n), \ldots, a_4 = (m, n), \ldots, a_7 = (m+1, n), a_8 = (m+1, n+1)$.

The feature map output from the score branch is input to the softmax function to select an action according to the score-based probability and use it for training. By passing the selected index to the box or mask branch, the bounding box that can express the position of the object at the corresponding index can be predicted.

2) STATE

The state S is defined as (7) in the form of a 2-tuple with the score map and the vector for the moving direction of the object.

$$S = (F_{score}, bb_d) \tag{7}$$

A score map is used to minimize the information appearing in the actual image. The direction of the object can be inferred using the bounding box estimated by the previously selected action. The unit vector for the movement direction extracted as the position of the bounding box is used. The movement direction from the previous 10 frames to the current frame is set in a vector form. Finally, in (7), F_{score} is the similarity score map, and bb_d is a vector containing the moving direction of the bounding box. Therefore, the state includes the location information of the part most similar to the target

3) REWARD

In most offline learning-based tracking algorithms, if tracking fails once within a sequence, then errors accumulate and drift to another target or background. It can be assumed that the tracking algorithm performs good job tracking when the tracking is successful in the last frame. Therefore, reward is defined through the IoU between the ground-truth bounding box (bb_G) of the last frame and the estimated bounding box (bb_E).

There are several ways to reward this work. For example, there are methods of comparing with ground-truth every frame, a method of giving the output score value as it is, and a method of giving a position difference between two boxes. However, if an overly accurate value is given as a reward using ground-truth, then the difference with supervised learning becomes ambiguous. The purpose of this study is to effectively learn in the section where an object occlusion occurs. To achieve this purpose, like ADNet, the reward is defined as in (8) so that if the IOU is 0.7 or more in the last

TABLE 1. Quantitative performance comparison on VOT2018.

	VOT2018				
	Reset Based				No Reset Based
Tracker	**Accuracy (A) ⇑**	**Robustness (R)⇓**	**EAO⇑**	**Speed(fps)⇑**	**AO ⇑**
SiamRPN	0.490	0.460	0.244	**200**	0.47
SiamRPN_R (Ours)	0.581	0.459	0.270	186.8	0.361
SiamMask	0.609	0.276	0.380	55	0.420
SiamMask_R (Ours)	**0.618**	0.290	0.390	32	0.383
SiamRPN++	0.600	0.234	0.414	75	0.495
SiamRPN++_R (Ours)	0.600	**0.234**	**0.416**	47.4	**0.508**

TABLE 2. Comparison of performance between the proposed model and the existing state-of-the-art model.

	VOT2018		
Tracker	**A ⇑**	**R⇓**	**EAO⇑**
DaSiamRPN	0.569	0.337	0.326
SA_Siam_R	0.566	0.258	0.337
CSRDCF	0.466	0.318	0.263
STRCF	0.523	0.215	0.345
DLSTpp	0.543	0.224	0.325
CPT	0.506	0.239	0.339
DRT	0.519	0.201	0.356
RCO	0.507	0.155	0.376
UPDT	0.536	0.184	0.378
MFT	0.505	**0.140**	0.385
LADCF	0.503	0.159	0.389
SiamFC	0.519	0.670	0.196
SiamRPN_R (Ours)	0.581	0.459	0.270
SiamMask_R (Ours)	**0.618**	0.290	0.390
SiamRPN++_R (Ours)	0.600	0.234	**0.416**

frame, a reward of 1 is given; otherwise, a penalty of -1 is given.

$$R_L = \begin{cases} 1, & \text{if } IoU = \frac{bb_G \cap bb_E}{bb_G \cup bb_E} > 0.7 \\ -1, & otherwise \end{cases} \tag{8}$$

4) STATE TRANSITION FUNCTION

After an action is selected in an arbitrary state, the state is changed to the next state, as shown in (9), by a state transition function based on the action.

$$s_{t+1} = f_{transition}(s_t, a_t) \tag{9}$$

If an action is selected in the current state, then the bounding box is estimated by the mask or box branch at the location of the action. After that, the similarity score of the next state is obtained. The next state is created by including the movement direction of the object obtained by the previously selected action in bb_d.

TABLE 3. Quantitative performance comparison on OTB50.

	OTB50	
Tracker	**Success ⇑**	**Precision ⇑**
SiamRPN++_R (Ours)	**0.673**	**0.881**
SiamMask_R (Ours)	0.627	0.866
SiamRPN++	0.665	0.875
SiamMask	0.617	0.840

C. IMPLEMENTATION

The reinforcement learning model is designed to extract the features of the score map by placing two 3×3 convolution layers and then output the nine previously defined actions through the fully connected layer. The average direction of 10 frames can be obtained as the average of the moving direction vectors (bb_d). Actions are selected by elementwise multiplication of the weights ($w_k, k \in [0, 8]$) for the average direction on the score map. It is expressed as (10), and an example is shown in Fig. 6.

$$a = M_{RL}(F_{score} \times w_k) \tag{10}$$

As shown in Fig. 6, if the average direction of bb_d is right, the right side (w_6) is given a higher weight than the remaining element values, and the average direction of the previous 10 frames in the state when selecting an action on the score map is considered.

The dynamic template exchange method uses the result of tracking with the highest score in the previous frame as a template when it is determined that tracking has failed. The template is correlated with the search image in a correlation layer with the target image. Here, the result of tracking with a higher score is used as the final output. Since the probability of missing the target is low at the beginning of tracking and there is no significant change in the object, it is designed to use the dynamic template exchange method after a certain frame interval.

D. TRAINING

The pretrained SiamMask model is used to output the score map during training. TrackingNet [52] is used for the dataset, and approximately 2,000 sequences, including

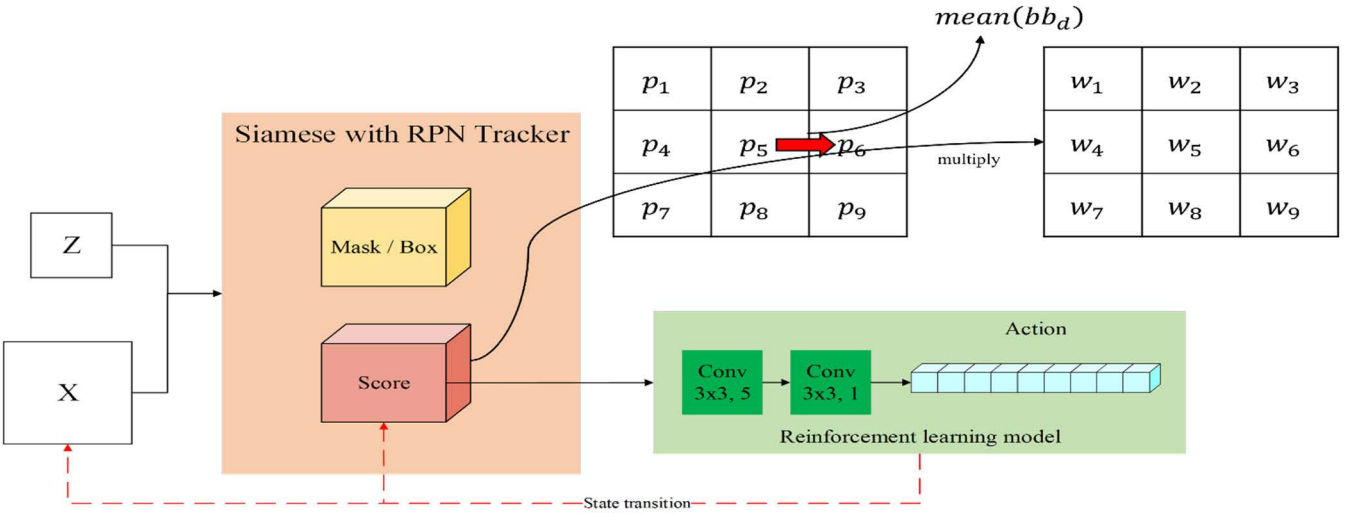

FIGURE 6. Example of giving weight to the average direction of the movement direction vector on the output score map when selecting an action.

object occlusion, are used for training. As mentioned earlier, Silver *et al.* [42] stated that the accuracy of reinforcement learning models can be improved through supervised learning. Therefore, before reinforcement learning, supervised learning is initially performed so that a more accurate action can be selected. First, for supervised learning, the data are customized so that the proposed model can be trained using a part of the training data. By inputting the sequence into SiamMask, a total of 9 inferences are performed on the adjacent index, including the max score index per frame. The score index with the highest overlap ratio in the last sequence is set to the same class as (6) as the ground-truth. If there are no significant factors that hinder tracking by supervised learning, the max score index is mostly selected as an action. However, when occlusion or motion blur occurs, the probability of selecting the max score index is drastically decreased. This situation has been experienced in reinforcement learning, and when an obstacle to tracking appears, an action that can succeed in tracking is selected.

Reinforcement learning models are trained by rewards obtained through actions in a specific environment state. In this research, the environment is set to a randomly selected sequence. The learning parameters are updated by the reward given in the last frame of the sequence. As mentioned earlier,

The reward is obtained in the last frame of the sequence belonging to the environment during training. Therefore, as in (11), the training parameters of the reinforcement learning model are updated using the SGA used in ADNet.

$$\Delta W_{M_{RL}} \propto \sum_{L}^{Env} \frac{\partial \log\left(p\left(M_{RL}\left(s_L\right)\right)\right)}{\partial W_{M_{RL}}} R_L \tag{11}$$

E. INFERENCE

Fig. 3 shows a flowchart of the inference process. First, the search image and target image are input to the Siamese network-based tracker, and the score map is input to the reinforcement learning model. The reinforcement learning model predicts the final bounding box by transferring the state of the score map and search image and delivering the selected index to the box/mask branch. At this time, when the output of the score map is less than 0.1, it is determined that the model has missed the target object. The tracked object in the frame with the highest score of the previous frame (f_p) is used as a dynamic template. Here, f_p is set to 10 in the same way as the number of action storage in Yun *et al.* [41]. We experimentally confirm that it takes approximately 3 frames when the tracker misses the object. Therefore, if it is set to a small number of 10 or less, there is a risk of performance degradation because there is a high possibility of using a template at the moment of missing a tracking object. When f_p is set to 5, the EAO of SiamMask_R decreases by approximately 0.03 by the VOT performance evaluation method. If f_p is set to be as high as 20, the tracking object within 10 frames after initialization is mainly used as a template. This drastically reduces the use of the target image initialized with the ground-truth, greatly increasing the number of missing objects. EAO is 0.04 lower when f_p is 20 compared to when f_p is 10.

We use a dynamic template after 50 frames. This is because it is assumed that the first 50 frames will be well tracked by initialization using ground-truth.

IV. EXPERIMENTS

In this section, experiments to verify the performance of the proposed algorithm and an analysis of the results are conducted. First, the experimental environment and the dataset used for performance evaluation will be described. Then, the experimental results are analyzed.

A. SETUP

The operating system of the experimental environment is Ubuntu 18.04, and the computer has the specifications of Intel i7-10700K CPU, Geforce RTX 2080 Ti (x2), and 32 GB of

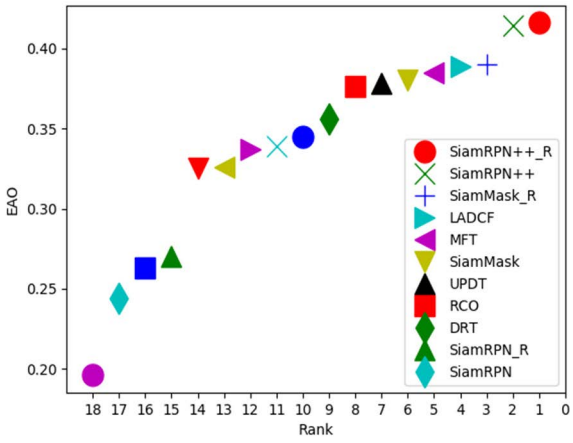

FIGURE 7. Expected average overlap rankings at VOT2018.

RAM. All proposed algorithms are written in Python, and PyTorch is used as the framework for deep learning.

B. DATASET

As a dataset for the objective quantitative evaluation of design methods, there are many benchmarks such as OTB50/100 [21], [60], VOT2016/2018/2019, LaSOT [61], and UAV123. However, the benchmark VOT2018, which has been used in many studies, and no-reset-based performance evaluation are used. OTB50, which can obtain various evaluation results, is adopted. VOT2018 was built with a total of 60 sequences considering many factors that interfere with tracking such as

illuminance, occlusion, motion, and scale, and the ground-truth was annotated with a rotated bounding box.

C. ANALYSIS OF RESULTS AND DISCUSSION

The models applying the proposed method to SiamRPN, SiamRPN++, and SiamMask are denoted as SiamRPN_R, SiamRPN++_R, and SiamMask_R, respectively.

1) VOT2018

In the VOT Challenge, the tracking algorithm is evaluated using accuracy, robustness, and EAO [62].

First, in Table 1, by applying the proposed framework to SiamRPN, SiamMask, and SiamRPN++, the performance before and after application is evaluated. Accuracy, robustness, and EAO are all adopted to compare performance. Additionally, to evaluate the one-pass evaluation (OPE), average overlap (AO) is adopted for performance comparison. In reset-based evaluation, performance is improved in all except robustness of SiamMask. Based on EAO, SiamRPN achieves a performance improvement of 2.6%, SiamMask 1%, and SiamRPN++ 0.2%. In the no reset-based evaluation, there is a performance improvement of 1.3% only in SiamRPN++, but the lowered performance is analyzed together with the qualitative results in Fig. 8.

Table 2 refers to the results of VOT2018. We compare our models with 12 state-of-the-art trackers [20], including DaSiamRPN [53], SA_Siam_R [54], CSRDCF [55], STRCF [56], DLSTpp [57], CPT, DRT, RCO, UPDT, MFT [58], LADCF [59], and SiamFC [9]. For accurate evaluation, the official VOT Toolkit is used, and the proposed framework is applied to SiamMask, SiamRPN, and SiamRPN++ and compared with 12 latest object trackers. When the proposed method is applied to SiamRPN++, as shown in Table 2, it surpasses the performance of all existing trackers, including the tracker evaluated with the highest rank in the VOT2018 Challenge based on EAO and accuracy. Compared with LADCF, which had won the challenge, it is 9.7% higher in accuracy and achieves a performance improvement of 2.7% in EAO. Although the EAO of SiamMask_R, which applies the proposed method to SiamMask, is lower than that of SiamRPN++, it shows higher performance than the existing tracking algorithm and has the highest accuracy. It achieves a performance improvement of 4.9% compared to DaSiamRPN, which achieves the best performance based on the existing accuracy.

In Table 1, all performance except for the robustness of SiamMask_R is improved in the reset-based evaluation. As seen from the book and helicopter sequence in Fig. 8, the bounding box of SiamMask_R is larger than that of other trackers. The reinforcement learning model passes the selected index to the mask branch to estimate the final bounding box based on the mask. At this time, if the index with a low score is selected, the mask includes the background, as shown in Fig. 5. The final bounding box is output large enough to include the background. Therefore, due to the accumulation of errors, the tracking fails and shows low robustness. However, it can be seen that the accuracy performance is improved by tracking more tightly to the ground-truth than the existing SiamMask in the frame in which the tracking is successful.

In the Flamingo1, soccer2, wiper, and helicopter sequences, it is confirmed that the reinforcement learning model robustly copes with occlusion by selecting an index different from SiamRPN and SiamRPN++.

Although the speed (fps) decreases due to the increase in computational cost by adding the framework, it still shows performance beyond real-time performance.

2) OTB50

The Object Tracking Benchmark (OTB) adopts success and precision to evaluate performance. Here, success is the overlap ratio between the tracking result and ground-truth, and it is the percentage of successful frames according to the threshold. Precision is an index indicating the percentage of the tracking result and the center distance of the ground-truth in pixels. In addition, we can check the performance of each attribute by evaluating the performance with the success performance index for 11 attributes. Performance is evaluated based on a one-pass evaluation. Hyperparameters are the same as those used in VOT2018.

First, as shown in Table 3, SiamMask_R shows performance improvement of 1% in success and 2.2% in precision compared to the existing model and performance

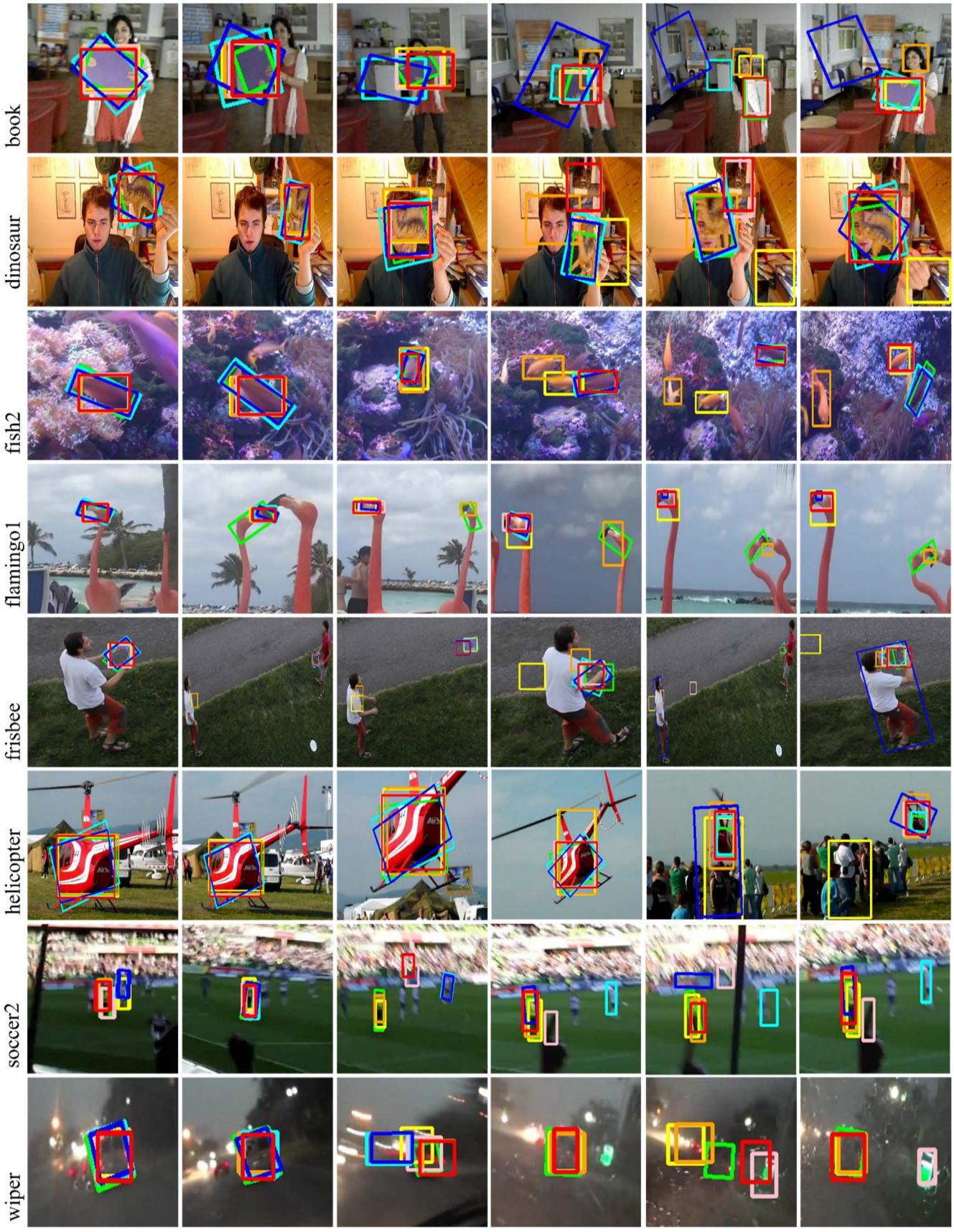

FIGURE 8. Qualitative results: We show some sample outputs on eight sequences selected from VOT2018, where the red box is SiamRPN++_R, the pink box is SiamRPN++, the blue box is siamMask_R, the cyan box is siamMask, the orange box is SiamRPN_R, the yellow box is SiamRPN, and the green box is the ground-truth.

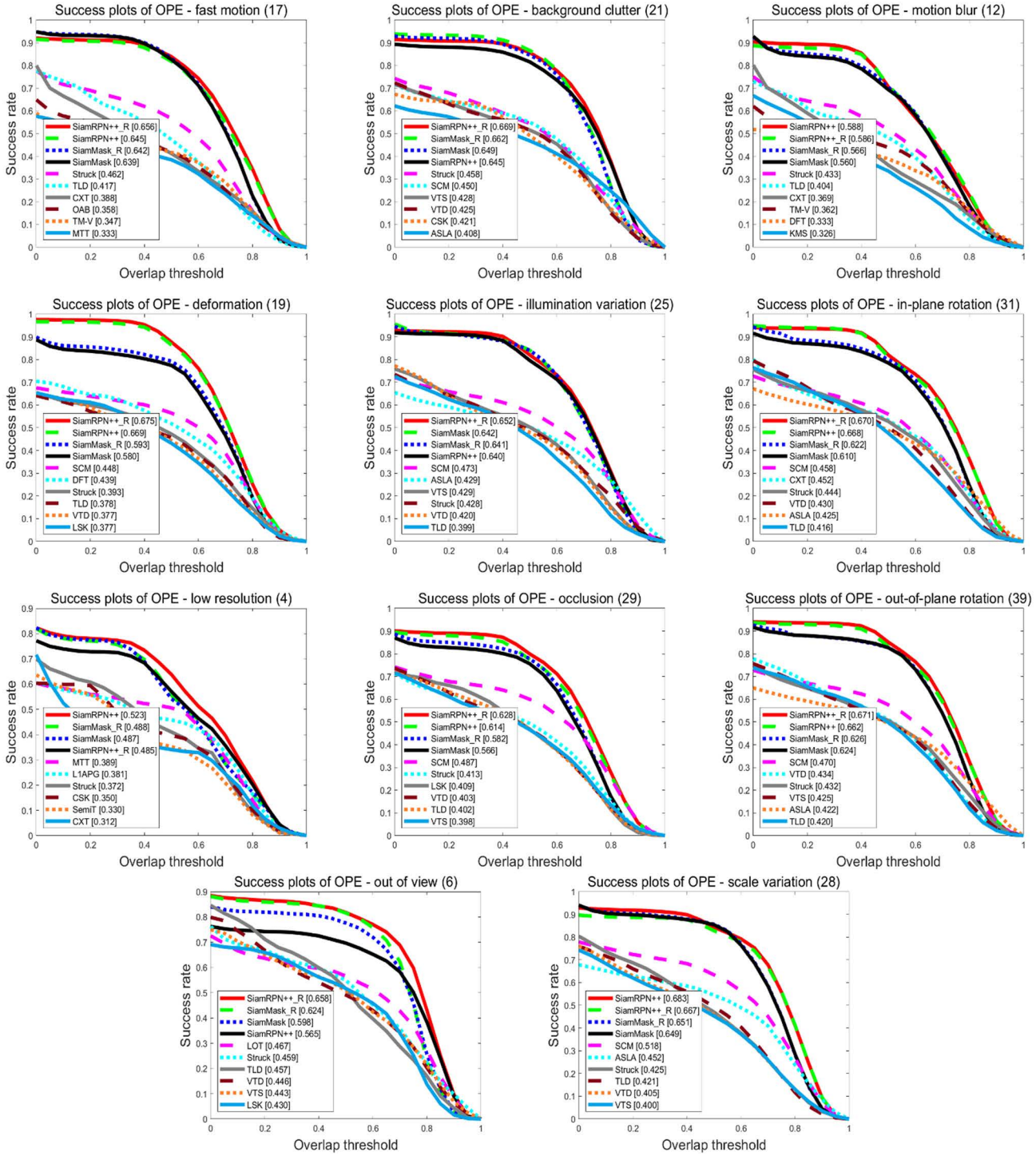

FIGURE 9. Quantitative results for OTB50 by 11 attributes.

improvement of 0.8% and 0.6% in SiamRPN++_R compared to the previous model. Additionally, as shown in Fig. 9, SiamRPN++_R using the proposed method shows the highest performance in all attributes except low resolution, motion blur, deformation, and scale variation. Although the performance of SiamRPN++_R in the above four attributes decreases, SiamMask_R shows higher performance than the existing SiamMask model. In particular, both of SiamRPN++_R and SiamMask_R show high performance in occlusion and out of view. This is because the reinforcement learning model selects the action to succeed in tracking when occlusion occurs in consideration of the moving

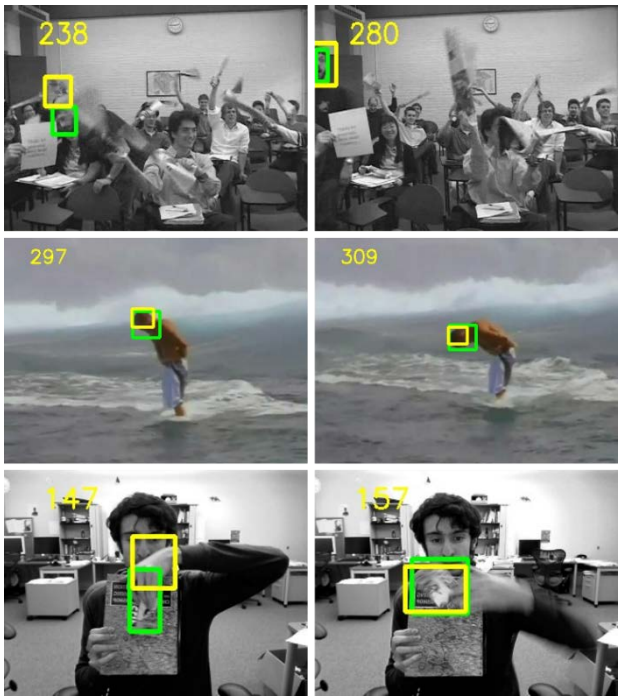

FIGURE 10. Example of performance degradation at low resolution and motion blur, where the green box is ground-truth and the yellow box is SiamRPN++_R

direction of the object and the score map. In the occlusion attributes, there is a performance improvement of 1.4% for SiamRPN++_R and 1.6% for SiamMask_R compared to the existing model. Because the dynamic template has the template of the frame tracked with the highest similarity recently, there is a substantial performance improvement of 9.3% for SiamRPN++_R and 2.6% for SiamMask_R compared to the existing model in the out-of-view attributes where the object disappears from view.

In Fig. 9, our method degrades the performance of SiamRPN++ for the low resolution and motion blur attributes. In Fig. 10, the first row is a sequence with low resolution attributes, the second row is a sequence with both low resolution and motion blur attributes, and the last row is a sequence with motion blur attributes.

Our method aims to successfully track the last frame of the sequence. In the sequence of low resolution attributes, we choose a location that completely misses the tracking object when the object moves quickly or when occlusion occurs. However, it continues to take an action to find the tracking object, and the tracking succeeds at the end of the sequence through the dynamic template. In the sequence of motion blur attributes, as shown in the third column of the frisbee sequence in Figure 8, the bounding box is predicted ahead of the object in the moving direction of the object. In the second row of Figure 10, we can see that the bounding box is visible at the end in the direction of the object's movement. In the third row, the upper part of the object is tracked, and the target object is tracked again even in the worst case of occlusion with other objects.

In the process of finding the target object again, it takes slightly longer for a sequence with motion blur and low resolution attributes than for a sequence in which a clear object appears. Therefore, a section with a low overlap with the ground-truth frequently occurs. Although the performance decreases in some sequences, the object is tracked again without reinitialization according to the design intention without the drift problem.

V. CONCLUSION

In this paper, we proposed a reinforcement learning model and dynamic template method to improve the performance of existing Siamese network-based trackers. Our proposed reinforcement learning models solve the occlusion problem by taking an action with a higher expected reward through experience of tracking successes and failures. The dynamic template exchange method prevents the drift problem by updating the template when the tracking model determines that the tracking object is lost. The proposed method outperforms existing state-of-the-art methods in VOT2018 and OTB50.

REFERENCES

[1] B. X. Chen, R. Sahdev, and J. K. Tsotsos, "Integrating stereo vision with a CNN tracker for a person-following robot," in *Proc. IEEE/CVF Conf. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 300–313.
[2] K. Koide and J. Miura, "Convolutional channel features-based person identification for person following robots," in *Proc. Int. Conf. Intell. Auton. Syst.*, Jun. 2018, pp. 186–198.
[3] N. Agarwal, C. W. Chiang, and A. Sharma, "A study on computer vision techniques for self-driving cars," in *Proc. Int. Conf. Frontier Comput.*, May 2018, pp. 629–634.
[4] A. Buyval, A. Gabdullin, R. Mustafin, and I. Shimchik, "Realtime vehicle and pedestrian tracking for didi udacity self-driving car challenge," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2064–2069.
[5] R. Xu, S. Y. Nikouei, Y. Chen, A. Polunchenko, S. Song, C. Deng, and T. R. Faughnan, "Real-time human objects tracking for smart surveillance at the edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[6] Y. G. Lee, T. Zheng, and J. N. Hwang, "Online-learning-based human tracking across non-overlapping cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, p. 2870–2883, Oct. 2017.
[7] S. You, H. Zhu, M. Li, and Y. Li, "A review of visual trackers and analysis of its application to mobile robot," 2019, *arXiv:1910.09761*.
[8] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 2, Lille, France, 2015, pp. 1–30.
[9] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, 2016, pp. 850–865.
[10] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.
[11] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.
[12] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.
[13] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," 2021, *arXiv:2103.17154*.

[14] H. Lee, S. Choi, and C. Kim, "A memory model based on the Siamese network for long-term tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, Sep. 2018, pp. 100–115.

[15] T. Yang and A. B. Chan, "Learning dynamic memory networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 152–167.

[16] A. Sauer, E. Aljalbout, and S. Haddadin, "Tracking holistic object representations," 2019, *arXiv:1907.12920*.

[17] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, Oct. 2016, pp. 749–765.

[18] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[19] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6182–6191.

[20] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. C. Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, and G. Fernandez, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 3–53.

[21] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. CVPR*, Portland, OR, USA, Jun. 2013, pp. 2411–2418.

[22] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, 2016, pp. 472–488.

[23] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[24] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE CVPR*, Jun. 2016, pp. 4293–4302.

[25] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. H. Lau, and M.-H. Yang, "VITAL: VIsual tracking via adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8990–8999.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2014, pp. 1–9.

[27] Y. Li and X. Zhang, "SiamVGG: Visual tracking using deeper Siamese networks," 2019, *arXiv:1902.02804*.

[28] I. Sosnovik, A. Moskalev, and A. Smeulders, "Scale equivariance improves Siamese tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2765–2774.

[29] Z. Zhang and H. Peng, "Deeper and wider Siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4600.

[30] X. Li, Q. Liu, N. Fan, Z. He, and H. Wang, "Hierarchical spatial-aware Siamese network for thermal infrared object tracking," *Knowl.-Based Syst.*, vol. 166, pp. 71–81, Feb. 2019.

[31] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan, and H. Wang, "Learning deep multi-level similarity for thermal infrared object tracking," *IEEE Trans. Multimedia*, vol. 23, pp. 2114–2126, 2021.

[32] Q. Liu, D. Yuan, N. Fan, P. Gao, X. Li, and Z. He, "Learning dual-level deep representation for thermal infrared tracking," *IEEE Trans. Multimedia*, early access, Jun. 6, 2022, doi: 10.1109/TMM.2022.3140929.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[34] B. X. Chen and J. K. Tsotsos, "Fast visual object tracking with rotated bounding boxes," 2019, *arXiv:1907.03892*.

[35] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6578–6588.

[36] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, Sep. 2014, pp. 740–755.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[38] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-VOS: Sequence-to-sequence video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2018, pp. 585–601.

[39] P. O. Pinheiro, R. Collobert, and P. Dollar, "Learning to refine object segments," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, Sep. 2015, pp. 75–91.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[41] 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. 2711–2720.

[42] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, and S. Dieleman, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[43] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.

[44] D. Zhang, H. Maei, X. Wang, and Y.-F. Wang, "Deep reinforcement learning for visual object tracking in videos," 2017, *arXiv:1701.08936*.

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

[46] M. Dunnhofer, N. Martinel, and C. Micheloni, "Tracking-by-trackers with a distilled and reinforced model," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Nov. 2020, pp. 631–650.

[47] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[48] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2014.

[49] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[50] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Cham, Switzerland: Springer, Sep. 2016, pp. 445–461.

[51] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J. K. Kamarainen, L. C. Zajc, O. Drbohlav, A. Lukezic, A. Berg, and A. Eldesokey, "The seventh visual object tracking VOT2019 challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2206–2241.

[52] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 310–327.

[53] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2018, pp. 101–117.

[54] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold Siamese network for real-time object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4834–4843.

[55] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4847–4856.

[56] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.

[57] L. Zheng, M. Tang, Y. Chen, J. Wang, and H. Lu, "Learning feature embeddings for discriminant model based tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 16. Glasgow, U.K., Aug. 2020, pp. 759–775.

[58] S. Bai, Z. He, Y. Dong, and H. Bai, "Multi-hierarchical independent correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2020, pp. 1–6.

[59] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, Nov. 2019.

[60] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[61] H. Fan, H. Ling, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, and C. Liao, "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5369–5378.
[62] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukezic, A. Garcia-Martin, A. Saffari, A. Petrosino, and A. S. Montero, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 564–586.

SUNG JUN PARK received the B.S. and M.S. degrees in electronics and information engineering from Korea Aerospace University (KAU), Goyang, South Korea, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronics and Information Engineering. His current research interests include image processing and computer vision.

SEUNG JUN HWANG received the B.S. and M.S. degrees in electronics and information engineering from the Korea Aerospace University (KAU), Goyang, South Korea, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronics and Information Engineering. His current research interests include image processing and computer vision.

JOONG-HWAN BAEK received the B.S. degree in telecommunication engineering from Korea Aerospace University, in 1981, and the M.S. and Ph.D. degrees from Oklahoma State University, in 1987 and 1991, respectively. He has been a Professor with School of Electronics and Information Engineering, Korea Aerospace University, since 1992. He was a Senior Researcher at the Electronics and Telecommunications Research Institute, South Korea, from 1991 to 1992. He is the Director of the Video-Audio Space Convergence Technology Research Center. His research interests include image processing, computer vision, pattern recognition, and multimedia.