Co-Inference Discriminative Tracking Through Multi-Task Siamese Network

YAN CHEN, JIXIANG DU, AND BINENG ZHONG

School of Computer Science and Technology, Huaqiao University, Xiamen 361021, China
Fujian Key Laboratory of Big Data Intelligence and Security, Huaqiao University, Xiamen 361021, China

Corresponding author: Yan Chen (yannya@hqu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61802135, Grant 61572205, and Grant 61972167; and in part by the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University under Grant ZQN-PY609.

ABSTRACT In essence, visual tracking is a matching problem without any prior information about a class-agnostic object. By leveraging large scale off-line training data, recent trackers based on Siamese networks usually expect to pre-learn underlying similarity functions before a tracking task even begins. Consequently, they lack discriminative and adaptive powers. To address the issues, we propose a multi-stage co-inference tracker (named MSCI) via a multi-task Siamese network, in which a complicated tracking task is divided into three complementary sub-tasks (i.e., classification, regression and detection). Firstly, we design a novel multi-task loss function to end-to-end train the multi-task Siamese network via jointly learning from three sub-tasks. The multi-task Siamese network contains three parallel yet collaborative output layers, which correspond to three key components of our tracker (i.e., classifier, regressor and residual learning based detector). By sharing representations within the components, we not only improve each component’s generalization performance, but also enhance our tracker’s discriminative power. Then, we design a co-inference approach to effectively fuse the complementary components. As a result, our tracker can avoid the pitfalls of purely single components and get reliably observations to improve its adaptive power. Comprehensive experiments on OTB2013, OTB2015 and VOT2016 validate the effectiveness and robustness of our MSCI tracker.

INDEX TERMS Visual tracking, multi-task learning, siamese network, residual learning, co-inference.

I. INTRODUCTION

Visual tracking is a fundamental task in computer vision, with a variety of applications including security surveillance, diverseness navigation, and robotics, etc. Given initial target states, a tracker will estimate the target states in successive video frames. However, it is greatly challenged by the time-varying target appearances caused by cluttered scenes, occlusions, and pose changes, etc.

To effectively capture complex and time-varying appearance changes, various top-performing trackers [1]–[6] have been proposed, such as on-line adaptive trackers [7]–[25], once-for-all pre-trained trackers [26]–[30], and hybrid trackers [31]–[35]. Although they have achieved promising performances from different aspects, they are challenged by different factors. First, on-line adaptive trackers are prone to model drift caused by using limited and noisy observations from the video itself for updating comparatively simple appearance models. Second, most once-for-all pre-trained trackers (e.g., GOTURN [28] and SiamFC [26]) lack the discriminative and adaptive powers because they pre-learn underlying similarity functions without updating. In addition, most of them use a simple substitution operation to update the target templates over time. Such an empirical substitution operation is unlikely to strike a good balance between model adaptivity and stability when there exist large appearance changes. Third, the hybrid trackers based on transforming a pre-trained convolutional neural network (CNN) suffer from scarcity of prior on a target object, and limited and corrupted training data. What’s more serious is that time consumption is an issue due to on-line updating the CNNs.

To address the above problems, we propose a multi-stage co-inference hybrid tracker (named MSCI) via a multi-task Siamese network. First, to address dilemma of real-time requirement and model complexity, we pre-train a multi-task Siamese network to capture underlying visual representations and similarity functions using solely large-scale off-line training samples. Without on-line training, the pre-trained multi-task Siamese network can run at a real-time speed during tracking. Second, to boost the discriminative
power of the proposed MSCI tracker, we divide a complicated tracking task into the classification, regression and detection sub-tasks. Meanwhile, we jointly train a classifier, regressor and detector from the output layers of the proposed multi-task Siamese network. By introducing reliable priors for regressing the target states and re-adjusting it, the classifier enables the proposed MSCI tracker to automatically perform object re-identification. Third, to capture large appearance changes and improve the adaptivity of our tracker, the residual learning based detector is jointly trained with the classifier and regressor via an end-to-end strategy. The classifier and regressor are expected to perform well most of the time, while achieve a real-time speed. The residual learning based detector does not work on every frame, but is only triggered once the classifier and regressor fail. Furthermore, based on the feedbacks from the residual learning based detector, the classifier and regressor may improve their tracking results by adjusting the target templates. Finally, by enforcing such a multi-stage co-inference approach, we not only boost our tracker’s discriminative power, but also greatly improve its adaptive power. In summary, the main contributions of this work are three-folds:

1. To the best of our knowledge, none of earlier studies have simultaneously divided the complicated tracking task into classification, regression and detection sub-tasks. By viewing visual tracking as a co-inference problem from multiple sub-tasks, we provide a new promising alternative framework to integrate the results of multiple different yet complementary models (i.e., classifier, regressor and detector). Therefore, previously or newly developed models can be potentially exploited by our framework.

2. We propose a multi-task Siamese network for visual tracking, in which a novel loss function is designed to jointly train three components (i.e., classifier, regressor and residual learning based detector) via an end-to-end strategy. Consequently, the proposed MSCI tracker not only benefits from the significantly improved discriminative representations, but also its real-time processing speed.

3. We design a multi-stage co-inference approach to combine three complementary components to address the dilemma of robustness and adaptivity of a tracker. The classifier and regressor from the multi-task Siamese network corporately cope with the gradual appearance changes. Meanwhile, the residual learning based detector effectively captures large appearance changes, and thus alleviates rapid model degradation caused by large appearance changes.

II. RELATED WORK

Please refer to [1]–[6] for more complete reviews on visual tracking which has been studied for decades. The trackers that are mostly related to our work can be roughly categorized into three classes: (1) on-line adaptive trackers that solely use on-line training samples to construct target appearance models; (2) once-for-all pre-trained trackers that solely use off-line training samples to pre-learn target appearance models; (3) hybrid trackers that simultaneously use on-line and off-line samples to learn appearance models.

A. ON-LINE ADAPTIVE TRACKERS

Traditional on-line adaptive trackers typically assume that they are unable to pre-learn the appearance models because collecting enough training data is luxury. Consequently, they use solely on-line training samples extracted from the video itself in a limited time period to learn a target appearance model. Some representative on-line trackers include on-line boosting based tracker [14], support vector machine based tracker [7], tracking-learning-detection [18], sparse coding based trackers [23]. However, these on-line trackers tend to drift when the target appearances keep changing and tracking errors are accumulated.

To alleviate the tracker drift problem, a huge amount of efforts have been applied in on-line trackers from different aspects, such as discriminative feature selection, classifier construction, training sample selection [36], occlusion detection [17], [24], multi-cue combination [13], [16], [19], [25], [37], and geometric restriction [10]. Recently, a variety of correlation filters based on-line trackers [8], [9], [11], [12], [15], [20], [21], [22], [38] have been proposed by efficiently and effectively using a large amount of negative samples. Nevertheless, these trackers heavily rely on the immediate target samples in a limited time period. Therefore, they may fail when the target appearances abruptly change to some previously unseen states.

B. ONCE-FOR-ALL PRE-TRAINED TRACKERS

The once-for-all pre-trained trackers can be roughly categorized into two classes: (1) with known categories, and (2) with unknown categories. The once-for-all pre-trained trackers with known categories [39]–[42] typically assume the categories of the targets are known before tracking. The assumption is so strong that the prior would be unavailable when the interested targets are unknown.

On the other hand, a variety of the once-for-all trackers via Siamese networks [26]–[30] resort to massive amounts of off-line training data to improve tracking performance. The trackers via Siamese networks construct a generic object tracker using solely large-scale off-line training data and without on-line training. Consequently, the trackers can run at a real-time speed during test time. Some representative trackers include GOTURN [28], SiamFC [26], YCNN [27], Re3 [43], and SINT [30]. Please refer to the survey paper [29] about various Siamese learning based trackers.

Although the trackers via Siamese networks (e.g., GOTURN and SiamFC) are fast at test time and have achieved good performance, they only contain a single output layer. And what’s worse are that they are sensitive to similar objects and the drifting problem due to lack of discriminative and adaptive power. In contrast, we divide the complicated tracking task into several simple yet effective sub-tasks in a multi-task Siamese network. Then, three complementary components corresponding to the sub-tasks are effectively combined.
C. HYBRID TRACKERS

Since both on-line adaptive and once-for-all pre-trained trackers have their strengths and weaknesses, there are some hybrid trackers aiming to perform simultaneous on-line and off-line training. Typically, hybrid trackers firstly use large scale off-line training samples to pre-train the feature representations. Then, with limited on-line training samples, the pre-trained feature representations are fine-tuned to a specific tracking task. Representative trackers include the stacked auto-encoder based tracker [35] and the CNN based trackers [31]–[34]. In the CNN based trackers, Hong et al. [31] and Li et al. [32] use the CNNs to pre-train hierarchical feature representations. In MDNet [33], a per-object classifier is on-line constructed based on an pre-trained image classification network. To exploit temporal information, Wang et al. [44] pre-train tracking features from auxiliary video sequences. Wang et al. [34] sequentially update the convolutional features. However, on-line transferring the pre-trained feature representations from deep models may be not only time-consuming, but also inhibiting tracking performance due to limited on-line training data.

In contrast, some authors propose to fix the pre-trained feature representations, while substitute hand-crafted features by them. One of the typically example is the correlation filters based trackers using CNNs. In [45], Martin et al. use pre-trained convolutional features to learn a discriminative correlation filter for visual tracking. Ma et al. [46] apply KCFs to several fixed and pre-trained CNN features. The resulting response maps are combined via a coarse-to-fine searching schema to effectively locate the target objects. Wang et al. [47] adaptively select and combined a variety of pre-trained CNN features for visual tracking. Qi et al. [48] view correlation filters based on pre-trained CNN features as weak trackers, and use an adaptive Hedge algorithm to boost the tracking performance. However, most of their trackers rely on ad hoc analysis to design the weights for different convolutional layers either using boosting or hedging techniques as an ensemble tracker. Consequently, some authors [49]–[51] explore how to make correlation filters based trackers using CNNs can benefit from the end-to-end training. Chen and Tao [52] use a single convolutional layer to learn a regression model for object tracking. Fan and Ling [53] propose a tracker which combines KCF based tracking and deep learning-based verification.

Based on SiamFC [26], Guo et al. [54] propose a dynamic Siamese network for visual tracking, in which a KCF model is firstly used to learn a fast transformation model. Then, the transformation model is used to update the pre-trained features in every frame. Although the dynamic Siamese network based tracker can achieve a nice balance between performance and speed, the on-line learned transformation models may be easily affected by the noisy and limited observations, and thus degrade the tracker. In contrast, all the components in our MSCI tracker are off-line trained via an end-to-end manner. In a word, most of the hybrid trackers still treat tracking and pre-training of feature representations as separated steps, and use conventional trackers without the help of the higher-level semantic feedbacks.

III. THE PROPOSED MSCI TRACKER

In this section, we firstly give an overview of the proposed MSCI tracker. Then, a novel multi-task Siamese network is designed to divide the complicated tracking task into a classification, regression and detection sub-task, respectively. Moreover, we develop a heuristic way to collect training data for the multi-task Siamese network. Finally, we present a residual learning based detector, which is used to enhance the stability and plasticity of the proposed MSCI tracker.

A. OVERVIEW OF THE PROPOSED MSCI TRACKER

Most trackers based on Siamese networks lack discriminative power because they overlook the discriminative information. Meanwhile, on-line adaptive trackers are sensitive to the drifting problem due to they are likely to use noisy or mis-aligned observations for model update. To improve our tracker’s discriminative power, while alleviating the drifting problem, we propose a multi-stage co-inference approach for state prediction. The flowchart of our MSCI tracker is schematically shown in Figure 1. The basic idea is to divide the complicated tracking task into a classification, regression and detection sub-task in the proposed multi-task Siamese network, respectively. As shown in Figure 1, our MSCI tracker is composed of three components: a classifier $C$, a regressor $R$, and a residual learning based detector $D$.

The classifier $C$ from the classification output layer of the multi-task Siamese network is used to calculate a probability that a candidate patch belongs to the positive class. In other words, it distinguishes that the candidate patch is a potential target region or not. The classifier enables the proposed MSCI tracker to automatically achieve object re-identification, by incorporating reliable prior into the regressor.

The regressor $R$ from the regression output layer of the multi-task Siamese network is responsible of predicting the position and scale offsets of a target object.

The residual learning based detector $D$ is used to increase the plasticity and stability of the proposed MSCI tracker.
when there exist large appearance changes. The residual learning is applied to take appearance changes into account via an end-to-end training strategy. The residual learning based detector is triggered once receiving a request from the classifier, and provides a feedback to the proposed MSCI tracker. As a result, the proposed MSCI tracker can effectively alleviate a rapid model degradation caused by large appearance changes. Meanwhile, the target templates of our proposed multi-task Siamese work are updated by reliable samples to capture the appearance changes.

Toward real-time and robust tracking, the three components work together via a multi-stage co-inference approach. Specifically, the proposed MSCI tracker works as follows. At an off-line training stage, by using the multi-task Siamese network, the classifier, regressor and detector are jointly trained on a classification, regression and detection sub-task, respectively. At an on-line tracking stage, a target template from the previous frame and a search region from the current frame are firstly fed into the classifier and regressor from the multi-task Siamese network, which is formulated by two-branch CNNs that share the parameters. Then, the classifier \( C \) calculates the probability of the candidate patch belonging to the positive class. Meanwhile, the regressor \( R \) estimates the candidate position and scale offsets of the target. Based on the obtained similarity from the classifier \( C \), the candidate patch is classified into three categories to trigger the co-inference approach to estimate the final target state, i.e., a positive sample with a high confidence, a positive sample with a low confidence, or a negative sample.

1. For a positive sample with a high confidence more than a threshold \( \gamma_h \), the output of the regressor \( R \) is utilized to directly estimate the position and scale offsets of the target.
2. Otherwise, for a positive sample with a low confidence within \([\gamma_l, \gamma_h]\), a set of dense candidate patches is firstly re-sampled according to previous states. Then, the probabilities of the re-sampled candidate patches belonging the positive class are calculated by the classifier \( C \). A candidate patch with the maximum probability to be a positive class is selected, and its output from the regressor \( R \) is used to predict the optimal target state.
3. In terms of the negative case, we argue that the output from the regressor \( R \) is unreliable due to significant appearance variations caused by rapid motion, image blur, low resolution and severe deformation, etc.

To cope with this issue, we introduce the residual learning based detector to collaboratively estimate the optimal target state. In the residual learning based detector, the residual learning is applied to effectively capture large appearance changes. As shown in Algorithm 1, our MSCI tracker iteratively continues until the end of the video. The remainder of this section will go into details of each component in the proposed MSCI tracker.

### B. THE PROPOSED MULTI-TASK SIAMESE NETWORK

Inspired by the success of multi-task learning in various applications, we propose a multi-stage co-inference approach to discriminative tracking via a multi-task Siamese network, in which the complicated tracking task is divided into a classification, regression and detection sub-task, respectively. As shown in Table 1, the proposed multi-task Siamese network consists of two convolutional input sub-branches and three parallel output layers. The two input sub-branches are constituted by a backbone CNN, in which the CNN parameters are shared to jointly extract robust features of input images, while reducing memory cost. The three parallel output layers are a classification, regression and detection layer, respectively. By sharing representations between three complementary tasks, we can not only enable our tracker to generalize better on its task, but also enhance its discriminative power. A novel multi-task loss function \( L \) is designed to end-to-end train the multi-task Siamese network:

\[
L = -\frac{1}{N_{cls}} \sum_i L_{cls}(p_i, u_i) + \lambda \frac{1}{N_{reg}} \sum_i u_i L_{reg}(b_i, g_i) + \beta \frac{1}{N_{det}} \sum_i L_{det}(y_i, l_i) \tag{1}
\]

where \( N_{cls}, N_{reg} \) and \( N_{det} \) are the number of training samples used for the classification, regression and detection sub-task, respectively. The \( L_{cls}, L_{reg} \) and \( L_{det} \) are the loss functions for the classification, regression and detection sub-task, respectively. The weighting hyper-parameters \( \lambda \) and \( \beta \) control the balance among the three sub-tasks. In the function \( L_{cls} \), the ground truth label \( u_i \) for \( i^{th} \) sample is set as 1 if it is a positive sample. Otherwise, it is set as 0. The \( p_i \) is the output probability for \( i^{th} \) sample from the classification layer.

### Algorithm 1 The Proposed MSCI Tracker

**Input:** An image sequence of length \( M \), an initialized bounding box of a target object, initialized parameters (i.e., \( \gamma_l, \gamma_h, \lambda, \) and \( \beta \)), a pre-trained multi-task Siamese network \( N \), a classifier \( C \) from \( N \), a regressor \( R \) from \( N \), a residual learning based detector \( D \) from \( N \).

**Output:** The optimal target state \( L_t \) at frame \( t \).

for \( t = 1 \) to \( M - 1 \) do

1. Crop a target template \( I_{t-1} \) from frame \( t - 1 \).
2. Crop a candidate patch \( I_t \) from frame \( t \).
3. Feed \( I_{t-1} \) and \( I_t \) into the network \( N \).
4. Calculate the probability \( Pr_t \) by the classifier \( C \).
5. Estimate candidate target state \( L^t_t \) by the regressor \( R \).

if \( Pr_t > \gamma_h \) then

- Set the optimal target state \( L_t \) as \( L^t_t \).
else if \( Pr_t \in [\gamma_l, \gamma_h] \) then

- Re-sample candidate patches.
- Calculate their probabilities to the target by \( C \).
- Select a patch \( p \) with the maximum probability.
- Set optimal state \( L_t \) as the output of \( R \) on \( p \).
else if \( Pr_t < \gamma_l \) then

- Predict optimal target state \( L_t \) via the detector \( D \).
end if

return the optimal target state \( L_t \) at frame \( t \).
end for
TABLE 1. The structure of proposed multi-task Siamese network. We abbreviate the fully connected and residual block to Fc and RB, respectively.

| Input Layer | Conv3-4 | Conv4-4 | Conv5-4 | Fc6 | Fc7 | Fc8 |
|-------------|---------|---------|---------|-----|-----|-----|
| Target Image | 224*224*3 | 56*56* | 28*28* | 14*14* | 1024 | 1024 | 1 |
| Search Image | 224*224*3 | 256 | 512 | 512 | 1024 | 1024 | 4 |
| Regression Layer | | | | | | |
| Fc6 | Fc7 | Fc8 |
| RB1 | 56*56*64 | | |
| RB2 | 28*28*64 | | |
| RB3 | 14*14*64 | 28*28 |

In the function $L_{reg}$, $b_i$ and $g_i$ are the bounding box regression offsets from the regression layer and ground truth label, respectively. In the function $L_{det}$, $y_i$ and $l_i$ are predicted and ground truth response maps, respectively. Note that only positive samples are used in the regression task.

One of the goals of the classification task is to improve the discriminative power of the multi-task Siamese network. Therefore, we utilize a cross entropy loss function for the binary classification task:

$$L_{cls}(p_i, u_i) = u_i \log p_i + (1 - u_i) \log(1 - p_i)$$

The goal of the regression task is to accurately estimate the position and scale offsets of a target object. Inspired by the work of fast r-cnn [55], we adopt a smooth $L_1$ loss function for the bounding box regression task to effectively handle the outliers and alleviate the gradient explosion:

$$L_{reg}(b_i, g_i) = \sum_{x \in \Delta x, \Delta y, \Delta w, \Delta h} Q_{L_1}(b_i^x, g_i^x)$$

$$Q_{L_1}(x) = \begin{cases} x^2 & \text{if } |x| < 1 \\ \frac{|x| - 1}{2} & \text{otherwise} \end{cases}$$

where $\Delta x, \Delta y, \Delta w, \Delta h$ are the position and scale offsets. The goal of the detection task is to capture the large appearance changes via residual learning with the target responses. The loss function of $L_{det}$ is formulated as the followings:

$$L_{det}(y_i, l_i) = ||y_i - l_i||^2$$

The detailed structure of the residual learning based detector and residual learning will be presented in Section III.D.

C. TRAINING DATA COLLECTION FOR MULTI-TASK SIAMESE NETWORK

To improve the effectiveness of our MSCI tracker, we propose a heuristic way to collect training data for the multi-task Siamese network. The multi-task Siamese network takes in two cropped images from a pair of successive frames as the input image pair, and returns three parallel output layers, i.e., a classification, regression and detection layer.

Figure 2 illustrates how training samples are drawn in a video sequence.

The first cropped image in the input image pair is a target template image which is cropped from a previous frame specified by the ground truth parameters. In addition, as shown in Figure 2(a), the size of first cropped image is typically extended to the twice size of the ground truth bounding box of an object. To generate a positive image pair for the classification layer, the second cropped image is obtained by extracting a warped image from the current frame specified by the ground truth parameters. Consequently, we have a positive image pair for the classification layer, and the classification label is set as 1. To get more positive and negative image pairs for training, we generate virtual data by Gaussian sampling on the current frame, and extract corresponding cropped images. As shown in Figure 2(b), our collection of positive data is heuristic based on the IoU between a sampled bounding box and ground-truth bounding box in the current frame. If the value of IoU is more than a threshold (typically set as 0.7), the image pair consisting of the sampled image and a target template image is a positive one, and its classification label is set as 1. Otherwise, it is a negative image pair, and its classification label is set as 0.

After collecting the training data for the classification layer, we need to design the regression labels for the regression layer in the multi-task Siamese network. The regression layer is responsible of predicting the position and scale offsets of a target. According to [28], a target object tends to smoothly move in the continuous frames. As shown in Figure 2(c), we assume that the position and scale offsets are subject to a Gaussian distribution. The relationship between the state of a target in the current frame and that in the previous frame can be formulated as:

$$x_c = x_p + w \cdot \Delta x$$

$$y_c = y_p + h \cdot \Delta y$$

$$w' = w \cdot \Delta w$$

$$h' = h \cdot \Delta h$$

where $(x_c, y_c)$ and $(w', h')$ denote the center coordinates and the scales of a target in the current frame, respectively. $(x_p, y_p)$...
and \((w, h)\) denote the center coordinates and the scales of a target in the previous frame, respectively. Consequently, the random variables \(\Delta x = \frac{y_j - y_i}{w} \) and \(\Delta y = \frac{y_j - y_i}{h}\) capture the position offsets. Meanwhile, the random variables \(\Delta x = \frac{y_j - y_i}{w} \) and \(\Delta h = \frac{y_j - y_i}{h}\) estimate the scale changes. Therefore, we use positive image pairs and their regression labels \((\Delta x, \Delta y, \Delta w, \Delta h)\) to train the regression layer.

To train the residual learning-based detector, we use a Gaussian function to generate a ground truth response map, which takes a value of 1 for a target position, and smoothly decays to 0 for the resting positions.

**D. THE RESIDUAL LEARNING BASED DETECTOR**

Once the classifier and regressor are unreliable, we introduce the residual learning-based detector to collaboratively predict the optimal target state.

Figure 3(a) shows the detailed structure of the residual learning in our detector. Firstly, we extract three-layer CNN features (i.e., Conv3-4, Conv4-4, and Conv5-4 features in VGG19 [56]) from the target and search images, respectively. Then, we apply the residual blocks (RB) [57] to capture large appearance changes. Finally, as shown in Table 1, we use the correlation operations to transform the RB features into the response maps:

\[
s_j = c_j(r_j(f_j(o_i))) \otimes r_j(f_j(o_s))
\]

where \(s_j\) is a response map denoting the similarity between the target image patch \(o_i\) and search image patch \(o_s\). \(\otimes\) is a correlation operation. \(f_j(\cdot)\) represents the \(j^{th}\) layer CNN feature of some properly trained CNN model, such as VGG19 [56]. \(c_j\) and \(r_j\) are a cropping operation and base residual mapping [57] applied on the \(j^{th}\) layer CNN feature, respectively. The base residual block adopted in this paper is shown in Figure 3(b). Please refer to [57] for more details. Finally, our expected detector output (i.e., an optimal response map) can be obtained by combining the response maps:

\[
y = \sum_j \lambda_j \times U_j(s_j)
\]

where \(\lambda_j \in [0, 1]\) is the weight of the \(j^{th}\) response map, and all the weights sum up to one. \(U_j\) is an interpolation operation, which makes sure the fused response maps have the same size. Finally, we adopt a scale pyramid based searching technique to handle the scale variations, and define the array of scaling factors as: \([0.98, 1, 1.02]\).

**IV. EXPERIMENTS**

In this section, the experimental settings and evaluation protocols are firstly described. Then, the comparisons between our MSCI tracker and state-of-art trackers on OTB2013 [4], OTB2015 [5] and VOT2016 [1] tracking benchmarks are systematically presented. Furthermore, the contribution of each component in our MSCI tracker is validated via an ablation study. Finally, we discuss the limitations of our MSCI tracker.

**A. IMPLEMENTATION DETAILS**

Experimental platform and computer configuration: The proposed MSCI tracker has been implemented by the Pytorch deep learning framework on the Pycharm platform. The running speed depends on the size of the search patch and video contents. Based on a HP Pro 3330 PC with Intel 2.4GHz CPU and GTX-980 GPU, the running speed of our MSCI tracker is about 27 fps at the resolution of 640×480 pixels.

How to train the proposed multi-task Siamese network: To obtain robust feature representations, we firstly utilize the pre-trained convolutional layers (i.e., Conv3-4, Conv4-4 and Conv5-4) from VGG19 [56] as the shared convolutional layers in the proposed multi-task Siamese network. Then, the shared convolutional layer parameters are fixed. Furthermore, the fully-connection layer parameters in both classification and regression layers are fine-tuned on the training data from the ALOV300+ [3], NUS-PRO [61] and ILSVC-2015 VID [58] video datasets. After removing seven overlapping videos with our testing set, there are still 307 video sequences containing 148,319 images in the ALOV300+ dataset. The NUS-PRO dataset contains 365 challenging video sequences covering 17 kinds of objects. In ILSVC-2015 VID, we only use 4,000 videos containing 1,300,000 RGB images. By setting the ratio of the number of negative samples to that of positive samples as 3, we obtain 120 input image pairs for the multi-task Siamese network from a pair of successive frames. The weight parameters \(\lambda_j\) for the three response maps \(s_j\) generated from Conv3-4, Conv4-4 and Conv5-4 are set to 0.3, 0.2 and 0.5, respectively. We apply a stochastic gradient descent (SGD) algorithm with 500,000 iterations to jointly optimize our network. The learning rate is initially set to 0.001, while its decay is set to 0.1 times every 100,000 iterations.

The parameter setting of the proposed MSCI tracker: Empirically, the parameters of \(\gamma_l\) and \(\gamma_h\) are set to 0.7 and 0.8, respectively. The weighting hyper-parameters \(\lambda\) and \(\beta\) in Equation (1) are set to 10 and 20, respectively.

**B. TRACKING BENCHMARK AND EVALUATION PROTOCOL**

To verify the performance of our MSCI tracker, we evaluate it on OTB2013 [4], OTB2015 [5] and VOT2016 [1] tracking benchmarks.

The OTB2013 and OTB2015 have 50 and 100 video sequences, respectively. In OTB2013 and OTB2015, the trackers are evaluated through two standard metrics,
FIGURE 4. The precision and success plots on OTB2013 [4] and OTB2015 [5] are shown in first and second row, respectively.

Typically, a distance threshold (usually 20 pixels) is used to determine if a tracker has successfully located a target or not in one frame. In the precision plots, the OTB2013 and OTB2015 rank the trackers based on the relative number of frames that the trackers have succeeded. On the other hand, the trackers can be ranked by the proportion of frames where the bounding box overlap exceeds a threshold (usually 0.5) in the success plots. For more details, please refer to the original papers on OTB2013 [4] and OTB2015 [5]. Different to the OTB2013 and OTB2015, the VOT2016 provides a re-initialized mechanism to reset target states once a tracker failed. In the VOT2016, the trackers are ranked based on a new measure called the expected average overlap (EAO). EAO is an estimator of the average overlap a tracker is expected to attain on a large collection of short-term sequences with the same visual properties as the given dataset.

C. EVALUATION ON OTB2013 AND OTB2015

Quantitative Evaluation: According to the evaluation protocol of the OTB2013 and OTB2015, we compare our MSCI tracker with several state-of-the-art trackers, e.g., GOTURN [28], KCF [15], CFNet-baseline-conv5 [50], SiamFC-3s [26], MEEM [25], MUSTer [16], SINT [30], HCFT [46], CREST [49], and DSiamM [54]. The compared trackers can be categorized into three types: (I) The on-line adaptive trackers, e.g., KCF, MEEM, and MUSTer; (II) The once-for-all pretrained trackers, e.g., GOTURN and SiamFC-3s; (III) The hybrid trackers, e.g., CFNet-baseline-conv5, SINT, HCFT, CREST, and DSiamM. The quantitative comparison results based on one-pass evaluation are shown in Figure 4. It can be seen from Figure 4 that our MSCI tracker is better than the state-of-the-art trackers. Due to leveraging the large scale off-line training data to learn robust feature representations, the proposed MSCI tracker has achieved better performance than on-line adaptive trackers, e.g., KCF, MEEM, and MUSTer. Compared to the results of GOTURN, our MSCI tracker observe improvements of 27.7% and 22.5% in the precision and AUC on OTB2013, respectively. Meanwhile, we also observe improvements of 28% and 21.2% in the precision and AUC on OTB2015, respectively. Compared to the results of SiamFC-3s, we observe significant improvements OTB2013 and OTB2015, respectively. The success of the proposed MSCI tracker lies in the multi-task Siamese network which divides the complicated tracking task into three complementary sub-tasks, and the multi-stage co-inference approach which can effectively address the dilemma between robustness and adaptivity of a tracker. Compared to The hybrid trackers (CFNet-baseline-conv5, SINT, HCFT, and CREST), the proposed MSCI tracker provides encouraging results. This is due to the proposed MSCI tracker can automatically perform object re-identification by systematically collaborating the adaptive KCF based tracker with the proposed multi-task Siamese network. Moreover, since DSiamM [54] only validates its tracker on OTB2013, we directly analyze the experimental results reported in the Figure 6 from its original paper [54]. DSiamM obtains 0.891 and 0.656 in the precision and AUC on OTB2013, respectively. However, our proposed MSCI tracker shows higher precision and AUC than the DSiamM by 2.1% and 2.3% on OTB2013, respectively. One of the main reasons is that the on-line learning transformation models adopted in DSiamM are easily affected by the noisy and limited observations. In contrast, we adopt the end-to-end strategy to learn the residual learning based detector which is more effective to capture the large appearance changes.

Qualitative evaluation: In Figure 5, we illustrate the qualitative comparison results of the competing trackers on six sequences, e.g., bird1, human3, diving, human9, dragonBaby, and motorRolling sequence. As we can see from Figure 5, the continuously and quickly foreground and background variations result in the failure of some trackers; while the targets can be accurately located by our MSCI tracker most these sequences. For example, the diving sequence suffers from deformation, scale variation, and in-of-plane rotation. The GOTURN, SiamFC-3s, MEEM, Muster, CREST, CFNet-baseline-conv5, and SINT have failed in the early stage due to the lack of either the adaptive or discriminative power. However, the proposed MSCI tracker has achieved robust
Y. Chen et al.: Co-Inference Discriminative Tracking Through Multi-Task Siamese Network

D. EVALUATION ON VOT2016
In this sub-section, we compare our MSCI tracker with state-of-the-art trackers on the VOT2016 dataset. The compared results on the EAO are shown in Figure 6. Since TCNN [59] use a complicated tree Structure to combine CNNs and bounding box regression based post-processing, its performance is a litter bit better than our MSCI. However, due to on-line updating the complicated tree Structure and CNNs, its average speed is 1.5 fps using MATLAB and a single NVIDIA GeForce GTX TITAN X GPU. Although the CCOT [60] is better than our MSCI on the VOT2016, its precision and AUC are 0.899 and 0.672 on the OTB2013, respectively. In contrast, the precision and AUC of our MSCI are 0.912 and 0.679 on the OTB2013, respectively. Overall, our MSCI gets comparable results in the VOT2016, while achieves a promising speed of 27 fps using Caffe and a single GTX-980 GPU. Please refer to [1] for more details on VOT2016.

E. ABLATION STUDY
Self-comparison: To verify each component’s contribution (i.e., the classifier, regressor, and residual learning based detector) in the proposed MSCI tracker, several variations of the proposed tracker are implemented and evaluated on the OTB2015. The first variation is denoted by “MSCI-M”, in which only the classifier and regressor is used for visual tracking, and the residual learning based detector is deleted from the original MSCI tracker. The second variation is denoted by “MSCI-Block”, in which the detector is trained without residual learning blocks. The third, fourth and fifth variation are denoted by “MSCI-Conv5”, “MSCI-Conv4” and “MSCI-Conv3”, in which the proposed MSCI tracker only use the Conv5-4, Conv4-4, and Conv3-4 CNN features from VGG19 to train its model (i.e., the three components), respectively. The sixth variation is denoted by “Residual Learning based Detector”, in which the purely detector component is used to track a target object. Please note that the GOTURN can be viewed as a baseline tracker which only contains a regressor. The self-comparison results are shown in Figure 7. Compared to the results of GOTURN, MSCI-M observes improvements of 21.2% and 14.5% in the precision and AUC on OTB2015, respectively. The reason is that MSCI-M jointly trains a classifier and regressor in the multi-task Siamese network by dividing a complicated tracking task into two simple yet complementary sub-tasks. By further incorporating a residual learning based detector into the multi-task Siamese network, the MSCI can use a multi-stage decision strategy to re-identification a target when the the classifier and regressor have failed. As a result, the MSCI not only performs better than MSCI-M, but also the purely residual learning based detector. Even without residual learning, the MSCI-Block outperforms the MSCI-M. This validates the effectiveness of the detector. Compared to the results of MSCI, the MSCI-Block without residual learning is degraded. In addition, the performance of MSCI is steadily improved by using multi-layer CNN features. This is consistent with our intuition that each component in the proposed MSCI tracker, i.e., the classifier, regressor, and residual learning based detector, is helpful to improve tracking performance.

Analysis on the running time: To the best of our knowledge, the comparison of the running time of different trackers is rather difficult. This is because not all trackers use the same hardware configurations and programming platforms. Consequently, we just use the reported speeds from the original papers in most cases. As shown in Table 2, we have achieved the processing speed of 27 fps, which is comparable to the speed of other trackers. Compared with GOTURN, the speed of MSCI-M is significantly reduced to 45 fps due to dense re-sampling in our multi-task Siamese network. Although the residual learning based detector is further added in our tracker, the speed of our tracker is not significantly affected. The reason is that the residual learning based detector only uses convolutional and correlation operations to efficiently verify the candidates. According to Figure 4 and Table 2, our MSCI has achieved state-of-the-art performance while keeping its computational demands as low as possible.

The residual learning based detector: when to trigger? One big advantage of our MSCI tracker lies in that the residual
TABLE 2. The comparison of running time between our MSCI, its variants, and state-of-the-art trackers.

| Tracker | GOTURN | MSCI\_M | MSCI\_Conv4 | MSCI\_Conv5 | MSCI\_Block | MSC\_I |
|---------|--------|---------|-------------|-------------|-------------|--------|
| FPS     | 165    | 45      | 39          | 40          | 29          | 27     |

| Tracker | GOTURN | Siamese-3s | SINT | CREST | DSiamese-3s | MSC\_I |
|---------|--------|-------------|------|-------|-------------|--------|
| FPS     | 165    | 86          | 4    | 1     | 25          | 27     |

TABLE 3. Precision for different configurations of $\gamma_l$ and $\gamma_h$ on the OTB2015.

| $\gamma_h$ | 0.5 | 0.6 | 0.7 |
|------------|-----|-----|-----|
| $\gamma_l$ |     |     |     |
| 0.6        | 0.786 |     |     |
| 0.7        | 0.799 | 0.824 |     |
| 0.8        | 0.782 | 0.837 | 0.841 |

FIGURE 8. Illustration of the triggering process on the residual learning based detector. Please see the texts for more details.

learning based detector can be “useful” when the classifier and regressor fail. To verify this advantage, we show the fusion process through three tracking examples and how the detector assists the MSCI to alleviate the tracker drift problem. In Figure 8, the first, second and third rows are from skiing, subway and matrix sequences in OTB2015, respectively. The first column shows the target positions with red bounding boxes in the previous frames. The second column visualizes some representative image patches and their scores provided by the classifier. The third column visualizes the response maps generated by the residual learning based detector. The fourth column shows the previous target positions (in the red bounding boxes), the predicted target positions from the classifier (in the blue bounding boxes), and the predicted target positions from the detector (in the yellow bounding boxes), respectively. As shown in Figure 8, the residual learning based detector may be triggered in a particular time when the output of the classifier is unreliable. For example, in the skiing sequence, the residual learning based detector is triggered when the target appearances are significantly changed caused by the non-rigid deformation. In the OTB2015 [5], the ratio of the frames triggering the detector to the whole frames is around 20%.

Parameter sensitive analysis: Since the confidence parameters $\gamma_l$ and $\gamma_h$ are two key parameters to trigger the fusing process, we analyze the performance of the proposed MSCI tracker as their values are changed. Specifically, we calculate the precisions on OTB2015 for different configurations of parameter values. Because of a huge amount of different combinations, we only examine several possible combinations of parameter values. The quantitative comparison results are shown in Table 3. According to the observations, for different combinations of parameter values, a good combination can be chosen across a wide range of parameter values. Moreover, it is easy to obtain a good combination of parameter values.

F. LIMITATIONS AND DISCUSSION

Although our MSCI tracker has achieved promising results compared with the state-of-the-art trackers, it has several failure cases as shown in Figure 9. Here we analyze the causes leading to the failure and provide the possible solutions. Firstly, our MSCI tracker fails to follow targets when sudden camera motion occurs, e.g., the couple sequence. The problem may be solved by enlarging the search window size. Secondly, our tracker cannot deal with the most challenging ironman sequence, which may be solved by introducing a robust updating strategy to alleviate noising samples to corrupt the appearance model.

V. CONCLUSION

In this paper, we proposed a co-inference algorithm for discriminative tracking via a multi-task Siamese network, which divides visual tracking into three sub-tasks, i.e., classification, regression and detection. By carefully collaborating a residual learning based detector with the classifier and the regressor in a multi-stage co-inference approach, our MSCI can make them work together to effectively address the dilemma of robustness and adaptivity of a tracker. Meanwhile, our MSCI has maintained a real-time processing speed. Based on comprehensive experiments on OTB2013, OTB2015 and VOT2016 dataset, our MSCI has demonstrated state-of-the-art performance in the recent literature.

REFERENCES

[1] M. Kristan, R. Bowden, and K. Lebeda, “The visual object tracking VOT2016 challenge results,” in Proc. Comput. Vis. Workshops (ECCV), Amsterdam, The Netherlands, Oct. 2016, pp. 777–823.
[2] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, “A survey of appearance models in visual object tracking.” ACM Trans. Intell. Syst. Technol., vol. 4, no. 4, pp. 1–48, Sep. 2013.
[3] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey.” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 7, pp. 1442–1468, Jul. 2014.
[4] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jan. 2013, pp. 2411–2418.
[5] 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.
[6] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Computing Surveys (CSUR), vol. 38, no. 4, p. 13, 2006.
[7] S. Avidan, "Support vector tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 8, pp. 1064–1072, Aug. 2004.
[8] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, " Staple: Complementary learners for real-time tracking," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1401–1409.
[9] Z. Chen, Z. Hong, and D. Tao, "An experimental survey on correlation filter-based tracking," 2015, arXiv:1509.05520. [Online]. Available: http://arxiv.org/abs/1509.05520
[10] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," IEEE Trans. Multimedia, vol. 19, no. 4, pp. 763–771, Apr. 2017.
[11] J. Fan, X. Shen, and Y. Wu, "What are we tracking: A unified approach of tracking and recognition," IEEE Trans. Image Process., vol. 22, no. 2, pp. 549–560, Feb. 2013.
[12] H. Grabner and H. Bischof, "On-line boosting and vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, 2006, pp. 260–267.
[13] 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. 2015.
[14] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 749–758.
[15] Y. Hua, K. Alahari, and C. Schmid, "Occlusion and motion reasoning for long-term tracking," in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2014, pp. 172–187.
[16] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[17] Z. Liu, Z. Lian, and Y. Li, "A novel adaptive kernel correlation filter tracker with multiple feature integration," in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2017, pp. 254–265.
[18] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional neural network," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4293–4302.
[19] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," Int. J. Comput. Vis., vol. 75, no. 2, pp. 247–266, Aug. 2007.
[20] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," IEEE Trans. Neural Netw., vol. 21, no. 10, pp. 1610–1623, Oct. 2010.
[21] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 10, pp. 1683–1698, Oct. 2008.
[22] Y. Gao, R. Ji, L. Zhang, and A. Hauptmann, "Symbiotic tracker ensemble toward a unified tracking framework," IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 4, pp. 695–709, Apr. 2014.
[23] Y. Chen, K. Alahari, and C. Schmid, "Occlusion and motion reasoning for long-term tracking," in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2014, pp. 194–209.
[24] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in Proc. Brit. Mach. Vis. Conf. Nottingham, U.K.: BMVA Press, Sep. 2014.
[25] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," IEEE Trans. Multimedia, vol. 19, no. 4, pp. 763–771, Apr. 2017.
[26] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 749–765.
[27] Y. Chen and W. Tao, "On the flow: A two-flow convolutional neural network for visual tracking," 2016, arXiv:1604.07507. [Online]. Available: http://arxiv.org/abs/1604.07507
[28] R. Pfugfelder, "An in-depth analysis of visual tracking with Siamese neural networks," 2017, arXiv:1707.00569. [Online]. Available: http://arxiv.org/abs/1707.00569
[29] R. Tao, E. Gavves, and A. W. M. Smulders, "Siamese instance search for tracking," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1420–1429.
[30] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in Proc. Int. Conf. Mach. Learn., 2015, pp. 597–606.
[31] H. Li, Y. Li, and F. Porikli, "Robust online visual tracking with a single convolutional neural network," in Proc. Asian Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 194–209.
[32] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4293–4302.
[33] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional neural network for visual tracking," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1373–1381.
[34] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 809–817.
[35] G. Li, Q. Huang, L. Qin, and S. Jiang, "SSOCBT: A robust semisupervised online CovBoost tracker that uses samples differently," IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 7, pp. 3611–3620, Jul. 2018.
[36] B. Liu, C. Gong, X. Huang, and T. Zhou, "Real: Real-time recurrent regression networks for visual tracking of generic objects," 2017, arXiv:1705.06368. [Online]. Available: http://arxiv.org/abs/1705.06368
[37] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4314–4322.
[54] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, “Learning dynamic siamese network for visual object tracking,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1–9.
[55] R. Girshick, “Fast R-CNN,” 2015, arXiv:1504.08083. [Online]. Available: http://arxiv.org/abs/1504.08083
[56] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556
[57] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, 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.
[59] H. Nam, M. Baek, and B. Han, “Modeling and propagating CNNs in a tree structure for visual tracking,” 2016, arXiv:1608.07242. [Online]. Available: http://arxiv.org/abs/1608.07242
[60] 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. Cham, Switzerland: Springer, 2016, pp. 472–488.
[61] A. Li, M. Lin, Y. Wu, M.-H. Yang, and S. Yan, “NUS-PRO: A new visual tracking challenge,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 335–349, Feb. 2016.

YAN CHEN received the B.S. and Ph.D. degrees in computer science from the University of Science and Technology (USTC), Hefei, China, in 2003 and 2010, respectively. She is currently an Associate Professor with the School of Computer Science and Technology, Huaqiao University, Xiamen, China. Her current research interests include pattern recognition, machine learning, and computer vision.

JIXIANG DU received the B.Sc. and M.Sc. degrees in vehicle engineering from the Hefei University of Technology, Hefei, China, in September 1999 and July 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, in December 2005. He is currently a Professor with the College of Computer Science and Technology, Huaqiao University, Quanzhou, China.

BINENG ZHONG received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2010, respectively. From 2007 to 2008, he was a Research Fellow with the Institute of Automation and the Institute of Computing Technology, Chinese Academy of Sciences. From 2017 to 2018, he was a Visiting Scholar with Northeastern University, Boston, MA, USA. He is currently a Professor with the School of Computer Science and Technology, Huaqiao University, Xiamen, China. His current research interests include pattern recognition, machine learning, and computer vision.

VOLUME 9, 2021

***