Siamese Local and Global Networks for Robust Face Tracking

Yuankai Qi, Shengping Zhang, Feng Jiang, *Member, IEEE*, Huiyu Zhou, Dacheng Tao, *Fellow, IEEE*, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Convolutional neural networks (CNNs) have achieved great success in several face-related tasks, such as face detection, alignment and recognition. As a fundamental problem in computer vision, face tracking plays a crucial role in various applications, such as video surveillance, human emotion detection and human-computer interaction. However, few CNN-based approaches are proposed for face (bounding box) tracking. In this article, we propose a face tracking method based on Siamese CNNs, which takes advantages of powerful representations of hierarchical CNN features learned from massive face images. The proposed method captures discriminative face information at both local and global levels. At the local level, representations for attribute patches (*i.e.*, eyes, nose and mouth) are learned to distinguish a face from another one, which are robust to pose changes and occlusions. At the global level, representations for each whole face are learned, which take into account the spatial relationships among local patches and facial characters, such as skin color and nevus. In addition, we build a new large-scale challenging face tracking dataset to evaluate face tracking methods and to facilitate the research forward in this field. Extensive experiments on the collected dataset demonstrate the effectiveness of our method in comparison to several state-of-the-art visual tracking methods.**

***Index Terms*— Face bounding box tracking, Local and global CNN representations, Correlation filter.**

I. Introduction

VISUAL tracking is a fundamental problem in computer vision, which is widely used in a large number of applications such as human emotion detection, video surveillanc, and human-computer interaction [1]–[5]. Different from facial landmark tracking [6], [7], which aims to localize landmarks on a face over time, face bounding box tracking in this article means to localize a tight axis-aligned bounding box of the face specified at the starting frame of a video. For simplicity, face bounding box tracking is referred to as face tracking from now on.

Although there exist many facial landmark tracking methods [7], [10]–[13], face tracking is still worthy of being explored further. First, most of facial landmark tracking methods just output landmarks of all the faces in an image

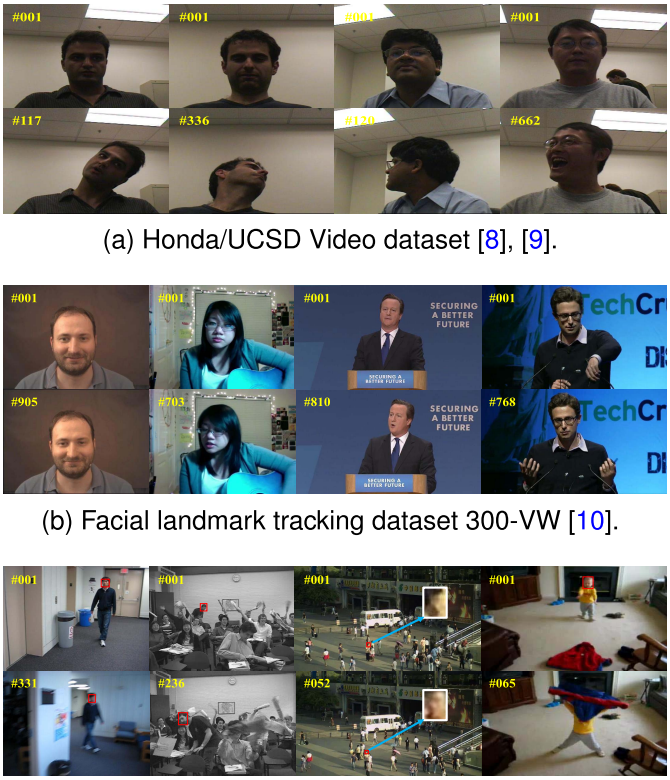

Fig. 1. Representative samples of the facial landmark tracking dataset 300-VW shown in (a), Honda/UCSD Video dataset shown in (b) and the face tracking dataset collected by ourselves shown in (c). Our dataset involves more real-world face tracking challenges, such as face blur, distractor faces, cluttered background, small faces, and heavy occlusions.

Manuscript received July 21, 2019; revised May 30, 2020 and August 16, 2020; accepted August 30, 2020. Date of publication September 17, 2020; date of current version September 29, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61902092 and Grant 61872112; in part by the National Key Research and Development Program of China under Grant 2018YFC0806802 and Grant 2018YFC0832105; and in part by the Fundamental Research Funds for the Central Universities under Grant HIT.NSRIF.2020005. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hichem Sahbi. *(Corresponding author: Shengping Zhang.)*

Yuankai Qi and Shengping Zhang are with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: yk.qi@hit.edu.cn; s.zhang@hit.edu.cn).

Feng Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: fjiang@hit.edu.cn).

Huiyu Zhou is with the School of Informatics, University of Leicester, Leicester LE1 7RH, U.K. (e-mail: hz143@leicester.ac.uk).

Dacheng Tao is with the Faculty of Engineering, School of Computer Science, The University of Sydney, Sydney, NSW 2008, Australia (e-mail: dacheng.tao@sydney.edu.au).

Xuelong Li is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China, and also with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: xuelong_li@nwpu.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.3023621

but do not distinguish one face from the others. Second, there is a lack of large-scale face tracking datasets. Existing face tracking methods carry out evaluations on a few of videos either selected from the OTB dataset [14] or collected by themselves [15]–[17]. As the number of test videos is usually less than fifteen, they cannot comprehensively reflect the performance of trackers. To the best of our knowledge, the current largest face tracking dataset is Honda/UCSD [8], [9], which contains 30 test videos captured indoor with clean backgrounds as shown in Figure 1a. Such videos cannot well reflect complicated challenges in real-world face tracking applications, such as video surveillance, where the background is usually cluttered and the face area could be much smaller. For the same reason, facial landmark tracking datasets, such as 300-VW [10] as shown in Figure 1b, are also not suitable for the face tracking evaluation. Third, CNN-based face tracking methods are far from widely explored. Although deep learning methods have achieved great success in a bunch of computer vision tasks (*e.g.*, image classification [18], image segmentation [19], object detection [20]) including face-related tasks such as face detection [21]–[34], recognition [35] and alignment [36], there are only a few CNN-based methods for face tracking [14], [17], [37], [38].

To make a step forward in the face tracking field, we collect a challenging large-scale dataset and propose a novel CNN-based face tracking method. In particular, the proposed tracking method is composed of two siamese CNNs: Local-CNN (**L-CNN**) and Global-CNN (**G-CNN**), which capture discriminative face information at both local and global levels. The L-CNN is designed to distinguish faces from a local patch level, which is responsible for capturing the differences between the corresponding patches of two faces, such as eyes, noses, and mouths. It is inspired by the facts that different persons are often characterized by the appearance differences in individual face constituent parts, *e.g.*, the shape/size/color of Person A's lips/nose/eyes is different from that of Person B's. On the other hand, solely modeling local patches is not always sufficiently distinguishable because (I) spatial relationships between local patches are important as well, such as the distance among eyes, nose and mouth, and (II) other crucial distinct factors may exist in the facial area outside the above mentioned local regions, such as skin color and nevus. Taking these into consideration, we further design a G-CNN to distinguish one face from the others from a global level. With representations obtained from L-CNN and G-CNN, we formulate face tracking in the form of ridge regression inspired by the success of correlation filters (CFs) in recent generic visual tracking [39]–[41]. In addition, we collect a dataset including a total of 50 videos, which are partially adopted from the OTB [1], VOT [2] and TC128 [42] datasets with manually re-annotated bounding boxes for each frame to adapt to the face tracking task. As shown in Figure 1c, the collected dataset covers both indoor and outdoor scenarios, and involves 11 kinds of tracking challenges, such as illumination changes, face blur caused by target fast motion or camera shaking, small faces, cluttered background, to name a few.

In summary, the contributions of this article are summarized below:

- We propose a CNN-based face tracking method by designing two kinds of siamese CNNs: L-CNN and G-CNN. The L-CNN distinguishes faces relying on information in local patches (*e.g.*, eyes, nose and mouth), while the G-CNN takes into account both spatial relationships between local patches and discriminative characters within the whole face.
- We build a large-scale challenging face tracking dataset, which reflects real-world scenarios. To the best of our knowledge, it is the largest dataset specially built for face tracking. We will make it publicly available to facilitate the research of face tracking.
- Extensive experiments on the collected dataset demonstrate the effectiveness of our method in comparison to several state-of-the-art visual tracking methods.

II. Related Work

In this section, we discuss the recent tracking methods that are closely related to our method.

A. Face Trackers

Most of the existing face tracking methods are based on hand-crafted features. McKenna and Gong [43] combine motion-based tracking and model-based face detection, which takes image intensities as input to produce face sequences from complicated scenes. Skin colors and facial shapes are utilized in [44], [45] to enhance tracking robustness. Yang and Paindavoine [46] implement an embedding face tracking system using a shallow neural network with only one hidden layer. Lee *et al*. [8] and Lee and Kriegman [9] propose a facial appearance manifold model to combine face tracking and face recognition. In [47], optical flow is adopted to calculate the likelihood of each pixel belonging to a face. Very recently, a few CNN-based face tracking methods have been proposed. Ren *et al*. [37] build two face detection CNNs with two different input spatial sizes. In [38], a CNN based on the pretrained AlexNet [48] is utilized to determine the head pose, which is used as a clue for face tracking. Intermediate CNN features are used as heat map in [14] to compute the location of a face according to a pre-defined threshold.

Instead of employing general image classification CNN architectures as mentioned above, we design three L-CNNs to learn representations of local face patches: eyes, nose and mouth, and a G-CNN to take into account spatial relationships between local face patches and distinct characters in the whole face.

B. CF and CNN Based General Trackers

In the past few years, correlation filters (CFs) based tracking methods attract increasing interests due to its computational efficiency. Dense sampling in these methods is approximated by generating a circulant matrix. Each row in the circulant matrix denotes a vectorized sample. With such kind of representation, a regression model can be efficiently solved in the Fourier domain. Bolme *et al*. [49] propose to learn robust filters by minimizing the sum of squared errors over image

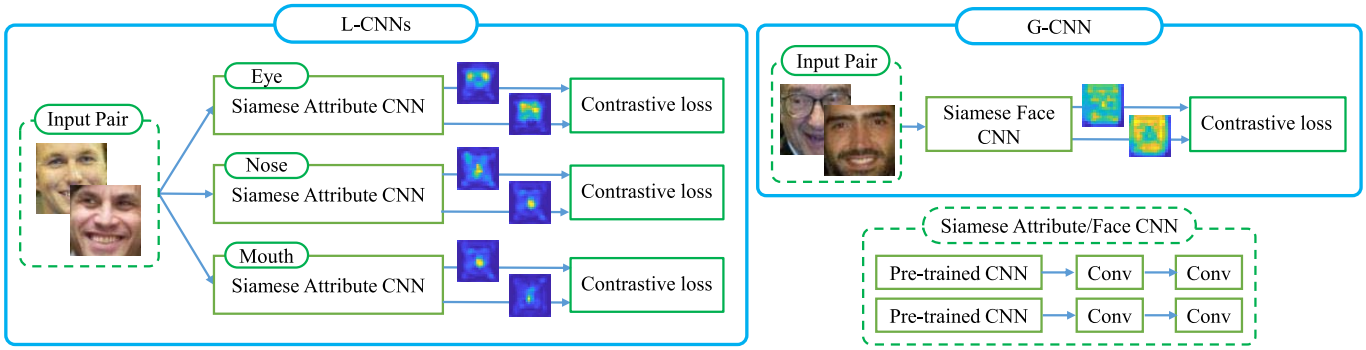

Fig. 2. The main architecture of the proposed L-CNN and G-CNN. The L-CNN includes three siamese attribute CNNs for the eyes, nose, and mouth, respectively. To guarantee each siamese attribute CNN predicts face similarity mainly relying on features of corresponding local facial region, we design a two-stage training strategy (see Section III-A). As a complementary, the G-CNN is designed to distinguish faces considering distinct information existing in the whole face, such as skin color and nevus, as well as spatial relationships among eyes, nose and mouth.

samples. Subsequently, Henriques *et al*. [50] extend single channel input features to multiple channels. At the same time, CNN-based tracking methods are developed in [51]. Nam and Han [52] propose a multi-domain CNN, where each domain is responsible for only one video so as to reduce the ambiguity caused by objects that could be the 'target' in some training videos but considered as 'background' in others. Yun *et al*. [53] introduce reinforcement learning into visual tracking, which formulates tracking as a series of actions on the target bounding box in the previous frame, such as translation moves and scale changes.

Lately, correlation filters are combined with multiple CNN features in [40] to take advantage of both low-level texture information and high-level semantic information. Danelljan *et al*. [39], [54] enable the integration of multi-resolution features with factorized correlation filters. Very recently, the correlation filter learning has been interpreted as a differentiable layer in a deep neural network in order to tightly couple feature learning with filter learning [55]. This method is further enhanced by introducing parallel IoU regression and binary classification [56], region proposal module [57], and deeper as well as wider architecture CNNs [3].

Unlike the above works, which aim at tracking general objects, in this work we specially focus on face tracking. This allows us to take advantage of prior knowledge of a face, such as face structure, during the design of our L-CNN and G-CNN.

III. Proposed Algorithm

In this section, we first present the proposed L-CNN and G-CNN to learn local and global facial representations. Then we build a tracker by learning correlation filters with the learned facial representations to take advantage of both the CF framework and the CNNs.

A. L-CNN

The proposed L-CNNs are composed of siamese attribute CNNs as illustrated in the left panel of Figure 2, which distinguish faces according to local regions of eyes, nose, and mouth, respectively. Each siamese attribute CNN consists of two parts. The first part is made up of five convolutional layers

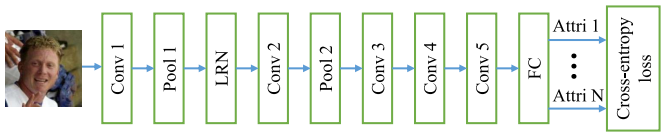

Fig. 3. Architecture of the eye/nose/mouth attribute classification network used in the first training stage of L-CNN.

adopted from AlexNet [48] considering both the complexity and the localizing ability as reported in [58] (other CNNs are also applicable). The second part is composed of two newly added convolutional layers, which are used to learn features facilitating to predict the similarity of two faces.

It is not straightforward to train L-CNNs because the borrowed layers from AlexNet are originally trained for generic object classification on the ImageNet [59] dataset and hence is not capable of focusing on the desired local facial regions. In addition, there is no dataset that comes up with pixel-wise annotations for facial local regions. To address this problem, we design a two-stage training method.

In the first training stage, we train the borrowed layers to locate local facial regions with the task of facial attribute classification on the CelebFaces Attributes (CelebA) dataset [60] inspired by [60], [61]. To this end, we append an FC layer to the rear as shown in Figure 3. We adopt the CelebA dataset as training data and pick eye-related, nose-related, and mouth-related attributes as classification labels. Specifically, the eye-related label set consists of *arched eyebrows, narrow eyes, bushy eyebrows, eyeglasses*, and *bags under eyes*; the nose-related label set consists of *big nose* and *pointy nose*; the mouth-related label set consists of *big lips, mouth sightly open, wearing lipstick,* and *smilling*. And hence, the FC layer of the eye/nose/mouth attribute classification network has 5, 2, and 4 outputs, respectively. Taking the eye attribute classification network as an example, and let $(X_i, y_i)_{i=1}^{M}$ be the training data, where $X_i \in \mathbb{R}^{P\times P}$ is a matrix denoting a face image and $y_i \in \mathbb{R}^{1\times Q}$ is a one-hot attribute label. The following cross-entropy loss is utilized for training

$$L_{attri} = -\sum_{i=1}^{M}\sum_{k=1}^{K} \mathbf{1}\{y_i == k\} \log \tilde{p}_i, \tag{1}$$

where K denotes the total number of sub-attributes, $\mathbf{1}\{y_i == k\} = 1$ if y_i is the k-th sub-attribute, otherwise $\mathbf{1}\{y_i == k\} = 0$, $\tilde{p}_i = p(y_i = 1|X_i)$ denotes the probability of the presence of one certain attribute using a sigmoid function.

In the second training stage, we remove the FC layer used in the first stage and add two convolutional layers at the rear to compose each branch of the siamese network as shown in the right bottom panel of Figure 2. In this stage, we train the newly added convolutional layers and fine-tune the layers trained in the first stage in order to better predict face similarity. In this stage, we utilize the cropped face images[1] of the LFW [62] dataset as the training data, which reduces background noise and therefore facilitates the network to predict face similarity mainly relying on the features of local facial regions. This dataset consists of ∼13,000 face images of 5,749 persons. To enable L-CNNs to recognize the face of a person in different scenes, we use the images of 1,680 persons who have at least two photos. The cross-entropy classification loss is also employed in this stage. Taking the eye siamese attribute CNN for example, we denote the input pair by (X_i, X_j, y_{ij}) where X_i and X_j are face images and $y_{ij} = 1$ if X_i and X_j are of the same person, otherwise $y_{ij} = 0$. Then the contrastive loss

$$L_{siam} = \sum_{i,j} y_{ij} d^2 + (1 - y_{ij}) max(0, margin - d)^2 \quad (2)$$

is adopted for training, where $d = \|\phi(X_i) - \phi(X_j)\|_F$, ϕ denotes the siamese network, and $margin$ denotes the desired minimal distance between two facial images.

B. G-CNN

Besides eyes, nose and mouse, the remaining facial region also contains crucial clues for face classification, such as face color and nevus. Furthermore, the spatial relationship between eyes, nose and mouth is also worthy of being considered. For these reasons, we design a G-CNN as a complementary network to L-CNNs. The G-CNN is equipped with a siamese architecture as illustrated in the top right and bottom right panels of Figure 2. It consists of three convolutional layers adopted from VGG-M [63] considering its classification accuracy and computation complexity (other CNNs are also applicable), and two newly added convolutional layers to better adapt to the task of face similarity prediction. The training process of G-CNN is the same as the second training stage of L-CNNs except the learning rate and the iteration parameters.

C. Tracking Under the CF Framework

Inspired by the success of state-of-the-art CF trackers [39], [54], here we adopt the continuous factorized CF technique of [39] to address the multi-resolution integration problem of the ridge regression. This technique allows us to take advantage of multi-resolution CNN features obtained by our L-CNNs and G-CNN.

For an input image region I, we first resize it to M different scales $\{I_1, \cdots, I_M\}$. Each image I_j is paired with a label map y_j, which is obtained by a two-dimension Gaussian function. Each element in y_j denotes the confidence of a pixel at its position in I_j being the center of the tracking target. For understanding convenience, notations similar to those of [39], [54] are adopted from now on. With the trained L-CNNs and G-CNN, for image sample I_j we extract outputs of L-CNNs and the G-CNN as its feature representation, denoted as x_j including D channels $x_j^1, \cdots, x_j^D$ and each channel may have different spatial resolutions $x_j^d \in \mathbb{R}^{M_d \times N_d}$.

Given an interpolation kernel, of which the period $T_1 > 0, T_2 > 0$, each feature map x_j^d is transferred to a continuous spatial domain $t_1 \in [0, T_1), t_2 \in [0, T_2)$ via

$$J_d\{x_j^d\}(t_1, t_2) = \sum_{n=0}^{N_d-1} \sum_{m=0}^{M_d-1} x_j^d[m, n] b_d(t_1 - \frac{T_1}{M_d} m) b_d(t_2 - \frac{T_2}{N_d} n), \quad (3)$$

where $J_d\{x_j^d\}$ is the interpolated feature and thus $J\{x\}$ denotes the whole interpolated features. Then filters are learned by minimizing the ridge regression in the contentious space

$$E(f) = \|S_f\{x_j\} - y_j\|_{L^2}^2 + \sum_{d=1}^{D} \|\omega f^d\|_{L^2}^2, \quad (4)$$

where $S_f\{x_j\} = f * J\{x_j\} = \sum_{d=1}^{D} f^d * J_d\{x_j^d\}$ is the correlation filter response (also called 'score' or 'target confidence'), the L^2-norm $\|g\|_{L^2}^2 = \frac{1}{T_1 T_2} \int_0^{T_2} \int_0^{T_1} |g(t_1, t_2)|^2 dt_1 dt_2$ and $\omega(t)$ is a spatial penalty function (the cosine function is adopted) to reduce the shortcomings of periodic assumption.

However, due to CNN features being usually of high dimensions, the filters f in object function Eq. (4) are accordingly with massive number of parameters. To reduce parameters, [39] transfers learning f to learning a small set of basis filters $f^1, \cdots, f^C$, $C < D$, where the original filter f^d is achieved as a linear combination of these basis filters $\sum_{c=1}^{C} p_{d,c} f^c$, and $p_{d,*}$ are learned coefficients. Let P denote all the coefficients with a size $D \times C$. Then we have the factorized version of the correlation filter response

$$S_{Pf}\{x_j\} = Pf * J\{x_j\} = \sum_{c,d} p_{d,c} f^c * J_d\{x_j^d\} = f * P^\top J\{x_j\}, \quad (5)$$

and the object function Eq. (4) turns to

$$E(f, P) = \|\hat{y}_j - \hat{z}_j^\top P \hat{f}\|_{\ell^2}^2 + \sum_{c=1}^{C} \|\hat{w} * \hat{f}^c\|_{\ell^2}^2 + \lambda \|P\|_F^2, \quad (6)$$

in the Fourier domain, where $\hat{g}$ of a (T_1, T_2)-periodic function g denotes the Fourier series coefficients $\hat{g}(k_1, k_2) = \frac{1}{T_1 T_2} \int_0^{T_2} \int_0^{T_1} g(t_1, t_2) e^{-i\frac{2\pi}{T_1} k_1 t_1} e^{-i\frac{2\pi}{T_2} k_2 t_1} dt_1 dt_2$, the ℓ^2-norm is defined by $\|\hat{g}\|_{\ell^2}^2 = \sum_{k_1,k_2} \left|\hat{g}[k_1, k_2]\right|^2$, $\hat{z}_j^d[k_1, k_2] = X_j^d[k_1, k_2] \hat{b}_d[k_1, k_2]$ be the Fourier coefficients of $z = J\{x\}$, and $X_j^d[k_1, k_2] = \sum_{n=0}^{N_d-1} \sum_{m=0}^{M_d-1} x_j^d[m, n] e^{-i\frac{2\pi}{M_d} m k_1} e^{-i\frac{2\pi}{N_d} n k_2}$, k_1 and $k_2 \in \mathbb{Z}$ is the discrete Fourier transform (DFT) of x^d.

By minimizing the classification loss of (6), the basis filters f and coefficient matrix P can be learned jointly. The

[1] http://conradsanderson.id.au/lfwcrop/

objective function (6) is a non-linear least squares problem and can be optimized by a conjugate gradient method with the usage of Gauss-Newton [64]. Use the first order Taylor series expansion to linearize the residuals in (6) correspondding to approximate the bilinear term $\hat{z}^\top P \hat{f}$ around the current estimate $(\hat{f}_i, P_i)$

$$\begin{aligned}\hat{z}_j^\top (P_i + \Delta P)(\hat{f}_i + \Delta \hat{f}) &\approx \hat{z}_j^\top P_i \hat{f}_{i,\Delta} + \hat{z}_j^\top \Delta P \hat{f}_i \\ &= \hat{z}_j^\top P_i \hat{f}_{i,\Delta} + (\hat{f}_i \otimes \hat{z}_j)^\top \mathrm{vec}(\Delta P),\end{aligned} \tag{7}$$

where $\hat{f}_{i,\Delta} = \hat{f}_i + \Delta \hat{f}$ and $\otimes$ denotes the Kronecker product. Substituting the first order approximation (7) into (6), we obtain the Gauss-Newton subproblem at the i-th iteration

$$\begin{aligned}\tilde{E}(\hat{f}_{i,\Delta}, \Delta P) = \|\hat{y}_j - \hat{z}_j^\top P_i \hat{f}_{i,\Delta} - (\hat{z}_j \otimes \hat{f}_i)^\top \mathrm{vec}(\Delta P)\|_{\ell^2}^2 \\ + \sum_{c=1}^{C} \|\hat{w} * \hat{f}_{i,\Delta}^c\|_{\ell^2}^2 + \mu \|P_i + \Delta P\|_F^2.\end{aligned} \tag{8}$$

The objective function (8) is a linear least squares problem as it constrains filter f to be with finite nonzero coefficients. The conjugate gradient method is employed to optimize (8) to obtain the filter $\hat{f}_{i,\Delta}^*$ and matrix increment ΔP^*, which are updated as $\hat{f}_{i+1} = \hat{f}_{i,\Delta}^*$ and $P_{i+1} = P_i + \Delta P^*$.

With the learned f and P, the target is determined by finding the maximum of the correlation filter response across all the image scales. The filters are updated every five frames by starting the above optimization process with the samples generated by a probabilistic generative model as reported in [39].

IV. Experimental Results

We present extensive evaluations of our **L**-CNN and **G**-CNN based **T**racking algorithm (LGT) in this section. First, we describe the evaluation protocols and the collected face tracking dataset. Then, we demonstrate the effectiveness of L-CNNs and the G-CNN with an ablation study. Finally, we report the experimental results of the proposed LGT algorithm against several state-of-the-art tracking methods.

Our tracker is implemented in MATLAB and the CNN models are trained using the MatConvNet toolbox [65] and Caffe [66]. L-CNNs are first trained on the CelebA [60] dataset for face attribute classification and then trained on LFW [62] for face similarity prediction. G-CNN is trained only on LFW for face similarity prediction. Correlation filters are learned using the first frame of each testing image sequence. The unoptimized implementation runs at about 4 FPS on a desktop with an Intel I7-4790k CPU, 16GB RAM, and a GTX 1080Ti GPU card. The input size for L-CNNs and the G-CNN are 250×250 and 224×224, respectively. The detailed architecture configurations are provided in Tables I and II.

TABLE I

Configuration of Each L-CNN. The ReLU Layer After Each Conv Layer Is Omitted for Clarity

	Kernel Size	Padding	Stride	Number of Kernels
Conv1	7x7	0	2	96
MaxPooling	3x3	0	3	–
LRN	Size: 5, Beta:0.75, Alpha: 0.0001			
Conv2	5x5	1	1	256
MaxPooling	2x2	0	2	–
Conv3	3x3	1	1	512
Conv4	3x3	1	1	512
Conv5	3x3	1	1	1024
For the 1st training Stage				
FC				
Cross-entropy Loss				
For the 2nd training Stage				
Conv6	3x3	1	1	512
Conv7	3x3	1	1	512
Flatten				
L2_Normalization				
Contrastive Loss				

TABLE II

Configuration of Each G-CNN. The ReLU Layer After Each Conv Layer Is Omitted for Clarity

	Kernel Size	Padding	Stride	Number of Kernels
Conv1	7x7	0	2	96
LRN	Size: 5, Beta:0.75, Alpha:0.0005			
MaXPooling	3x3	0	2	–
Conv2	5x5	1	2	256
LRN	Size: 5, Beta:0.75, Alpha:0.0005			
MaxPooling	3x3	0	2	–
Conv3	3x3	0	1	512
Conv4	3x3	0	1	512
Conv5	3x3	0	1	512
Flatten				
L2_Normalization				
Contrastive Loss				

A. Evaluation Protocols

Following the standard paradigm in visual tracking, all the trackers are evaluated using precision and success plots of one-pass evaluation (OPE) [1]. The precision in precision plots is the ratio of frames that are with the Euclidean distance (between the center points of the ground truth and the tracking results) smaller than a given threshold in pixels. The success rate in success plots is the proportion of successful frames that have an Intersection over Union (IoU) no smaller than a threshold. Trackers in precision plots are ranked according to the precision at the threshold of 20 pixels following the popular tracking benchmark [1]. As a compensation to the fix threshold ranking in precision plots, the Area Under the success rate Curve (AUC) is used in success plots to rank the trackers, which is an integral on the whole range of success rates.

Fig. 4. Representative challenging samples of the collected face tracking dataset, which are challenged by full/partial occlusion, rotation, heavy illumination variation, face blur, and cluttered background.

B. Collected Dataset

We collect and manually annotate a new challenging dataset for evaluating face tracking methods. To the best of our knowledge, this is the largest face tracking dataset involving indoor and outdoor scenarios. The dataset consists of 50 re-annotated videos selected from three visual tracking datasets: OTB100 [1], TC128 [42] and VOT2017 [2]. Representative examples of the collected dataset are shown in Figure 4, which involves 11 kinds of challenges, such as huge illumination variance, face blur, cluttered background, small faces and heavy occlusions. In Table IV, we present the detailed challenges and resolution of each image sequence. The bounding box is annotated to cover the area from chin to forehead but not include the ears. Figure 5 shows annotating examples for front and side views. The dataset will be made publicly available.

The collected dataset is extremely challenging. We test three recently published state-of-the-art general object tracking methods: SiamRCNN [57], SiamDW [3], and ATOM [56], on our dataset. Table III presents the performance in terms of the main metric AUC. A significant performance drop of about 20% is observed.

Fig. 5. Bounding box annotation samples for front and side views.

TABLE III
AUC RESULTS OF STATE-OF-THE-ART GENERAL OBJECT TRACKING METHODS ON THE WIDELY USED GENERAL OBJECT TRACKING DATASET OTB100 AND OUR FACETRACKING DATASET

	SiamRCNN [59]	SiamDW [3]	ATOM [58]
OTB100	0.70	0.67	0.66
FaceTracking	0.53	0.42	0.47

C. Ablation Analysis

The proposed method uses features extracted by both L-CNN and G-CNN. L-CNN contains three siamese attribute CNNs for eyes, nose, and mouth, respectively. To evaluate the contribution of each siamese attribute CNN, we conduct an ablation analysis by comparing the tracking performance of the proposed method with the alternative method that uses

TABLE IV

VIDEO RESOLUTION AND CHALLENGES. 'IV': ILLUMINATION VARIATION, 'SV': SCALE VARIATION, 'OCC': OCCLUSION, 'DEF': DEFORMATION, 'FM': FAST MOTION, 'MB': MOTION BLUR, 'IPR': IN-PLANE ROTATION, 'OPR': OUT-PLANE ROTATION, 'OV': OUT-OF-VIEW, 'BC': BACKGROUND CLUTTERS, 'SF': SMALL FACE

ImageSequence	Resolution	Challenges	ImageSequence	Resolution	Challenges
BlurBody	640x480	SV, DEF, MB, FM, IPR, OPR	Skater	320x240	SV, DEF, IPR, OPR, OCC, SF
Boy	640x480	SV, MB, FM, IPR, OPR	Soccer	640x360	IV, SV, OCC, MB, FM, IPR, OPR, BC
ClifBar	320x240	SV, OCC, FM	Surfer	480x360	SV, FM, IPR, OPR, SF
Dancer2	320x262	DEF, OCC, SF, OPR, IPR	Trellis	320x240	IV, SV, IPR, OPR, BC
David	320x240	IV, FM, MB, OPR	Badminton_ce2	1280x720	DEF, MB, OPR, IPR, FM, OCC
David2	320x240	IPR, OPR	Ball_ce1	864x480	OCC, MB, IPR, OPR, FM
David3	640x480	OCC, DEF, OPR, BC, SF	Ball_ce4	640x480	FM, OPR, IPR, MB
DragonBaby	640x360	SV, OCC, MB, FM, IPR, OPR, OV	Busstation_ce2	1280x720	SF, OPR, BC
Dudek	720x480	SV, OCC, DEF, FM, IPR, OPR, OV, BC	Cup_ce	864x480	OPR
FaceOcc1	352x288	OCC	Face_ce	1280x720	SV, OCC, IPR, OPR, BC
FaceOcc2	320x240	IV, OCC, IPR, OPR	Face_ce2	640x360	IV, OCC, MB, IPR, OPR, FM
FleetFace	720x480	SV, DEF, MB, FM, IPR, OPR	Hand_ce2	640x480	OPR, IPR
Freeman1	360x240	SV, IPR, OPR	Hurdle_ce1	1280x720	DEF, MB, BC, SF, FM, OCC, OPR
Freeman3	360x240	SV, IPR, OPR	Hurdle_ce2	960x720	DEF, FM, BC, SF, MB, OCC, OPR
Freeman4	360x240	SV, OCC, IPR, OPR	Microphone_ce2	640x480	OPR
Girl	128x96	SV, OCC, IPR, OPR	Singer_ce1	448x336	IV, DEF, FM, BC, MB, OCC
Gym	426x234	SV, DEF, IPR, OPR, SF, FM, MB, OCC	Singer_ce2	864x480	IV, SV, DEF, OPR, BC
Human2	480x640	IV, SV, MB, OPR, FM	Skating_ce1	864x480	SV, OCC, DEF, FM, IPR, OPR, MB
Jumping	352x288	MB, FM	Skating_ce2	1280x720	SV, MB, FM, IPR, OPR, OCC
KiteSurf	480x270	IV, OCC, IPR, OPR	Sunshade	352x288	IV
Man	241x193	IV	Yo-yos_ce3	1280x720	OPR, OCC
Mhyang	320x240	IV, DEF, OPR, BC	Blanket	320x240	OCC, SV
Shaking	624x352	IV, SV, IPR, OPR, BC	Book	640x480	IV, FM, OCC, SV
Singer1	624x352	IV, SV, OCC, OPR	Dinosaur	320x240	OCC
Singer2	624x352	IV, DEF, IPR, OPR, BC, OCC	Hand	320x240	OCC, IPR

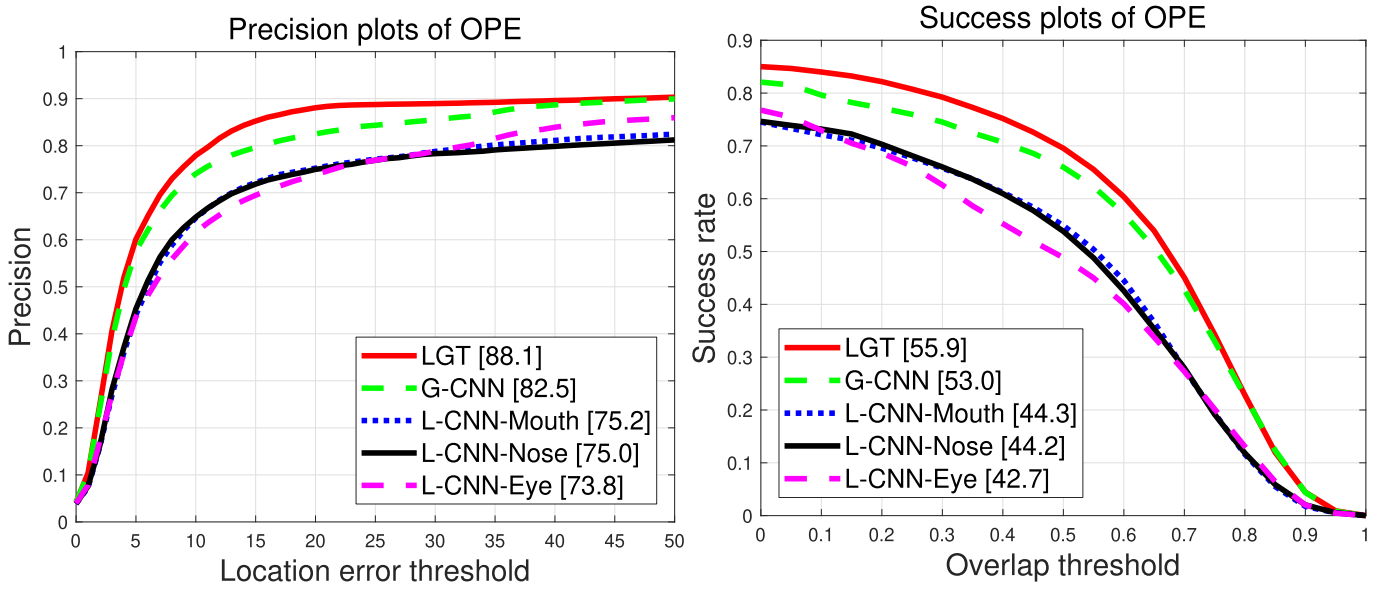

Fig. 6. Tracking results obtained by using local CNN features (*i.e.*, L-CNN-Nose/Mouth/Eye), global CNN features (G-CNN), and their combination (LGT).

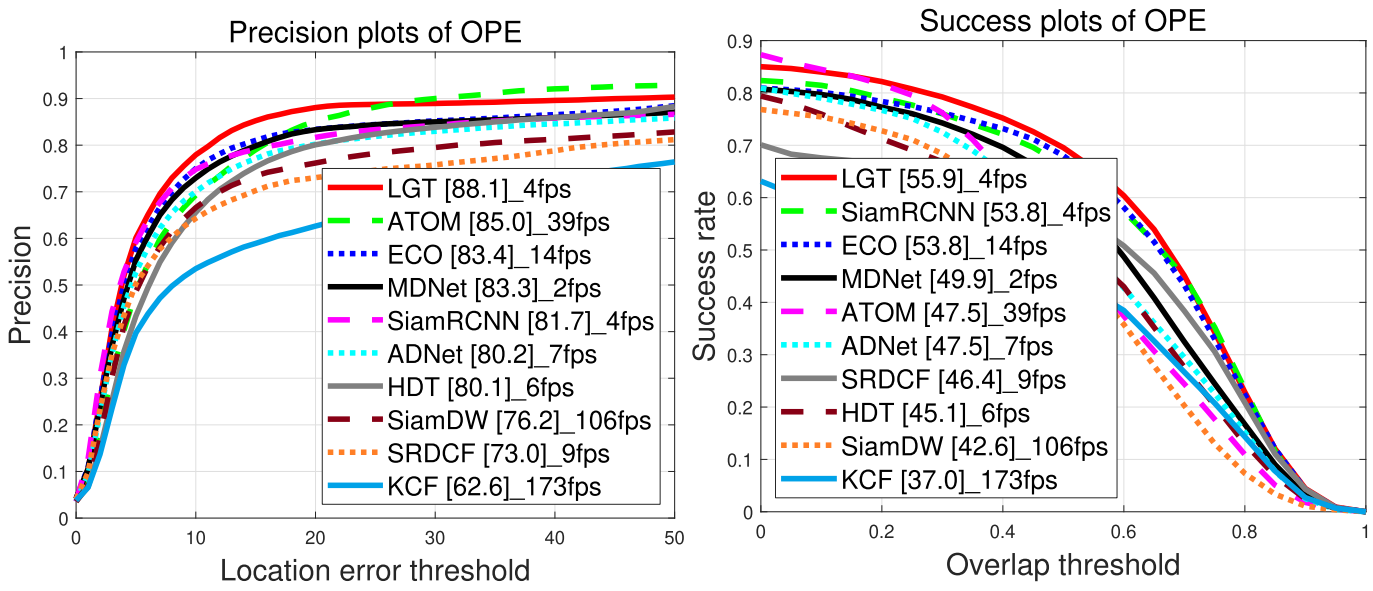

Fig. 7. Precision plots and success plots on the collected face tracking dataset compared against several state-of-the-art tracking methods.

TABLE V
COMPARISONS AGAINST FACE REPRESENTATION OBTAINED BY VGG-FACE [67] AND DEEPID [68]

	Ours_VGGFace	Ours_DeepID	Ours
Prec.	87.1	81.7	88.1
AUC	54.7	51.2	55.9

only G-CNN and the alternative methods that use one attribute CNN of eyes, nose or mouth, respectively.

The tracking results are shown in Figure 6 where the proposed method achieves the best tracking results compared to any of its components by a large margin of about 5% in terms of the precision plots and 3% in terms of the success plots. In addition, G-CNN contributes about 10% more to the tracking performance than the three L-CNNs. Among all the three attribute L-CNNs, the features of mouth regions play a slightly more important role than that of nose and eyes. The sole eye attribute CNN performs the worst, which may be caused by the fact that the eye region is very small and hence carries little information.

To further evaluate the effectiveness of the learned representations by the proposed L-CNNs and G-CNN, we test the performance by replacing L-CNNs and G-CNN with VGG-Face [67] and DeepID [68]. These two deep CNNs are trained by their authors for face recognition using a larger dataset than the data we used to train L-CNN and G-CNN. The results are presented in Table V. The results show that represetations obtained by our network achieve the best performance.

D. Comparisons to State-of-the-Art Trackers

We evaluate the proposed LGT algorithm against eight state-of-the-art tracking methods including seven recently published CNN-based trackers (SiamRCNN [57], SiamDW [3], ATOM [56], ECO [39], MDNet [52], ADNet [53], CFNet [55], HDT [40]) and two hand-crafted feature based trackers (KCF [50] and SRDCF [69]). All the trackers are not trained on the proposed dataset. Below we present a brief introduction of each tracker and the training data.

KCF is a correlation filter based tracker using HoG [70] feature; SRDCF introduces a spatial regularization into KCF; HDT integrates multiple CNN features into KCF using an adaptive Hedging algorithm (VGG-19 pretrained on ImageNet [59] is employed to extract CNN features); ECO integrates multi-resolution CNN features with factorized correlation filters (VGG-M [63] pretrained on ImageNet is employed to extract CNN features); CFNet formulates correlation filter using learnable layers in a siamese network (trained using ILSVRC15-VID [71]); ADNet formulates visual tracking as a series of translation and scaling operations of the target in the previous frame (trained on VOT13∼VOT15 [72]–[74] and ALOV300 [75]); MDNet formulates visual tracking using a multi-domain binary classification CNN (trained on VOT13∼VOT15 [72]–[74]); ATOM employs two parallel networks: one for target/background classification, and the other one for IoU prediction (trained on LaSOT [76] and TrackingNet [77]); SiamRCNN and SiamDW are based on siamese networks simultaneously implementing correlation filtering and object proposing. SiamRCNN emphasizes re-detection on previous frames and SiamDW focuses on a deeper and wider network architecture. SiamRCNN is trained on LaSOT [76], ILSVRC15-VID [71], YouTube-VOS2018 [78], and GOT-10k [79]. SiamDW is trained on YouTube-BB [80] and ILSVRC15-VID [71]. In contrast, our model is trained on CelebA [60] and LFW [62] for face attribute classification and face similarity prediction.

1) Quantitative Evaluation: Figure 7 shows the tracking results on the collected face tracking dataset in terms of success plots and precision plots. The results show that the proposed LGT method achieves the best performance in terms of both metrics, while the runner-ups are three different

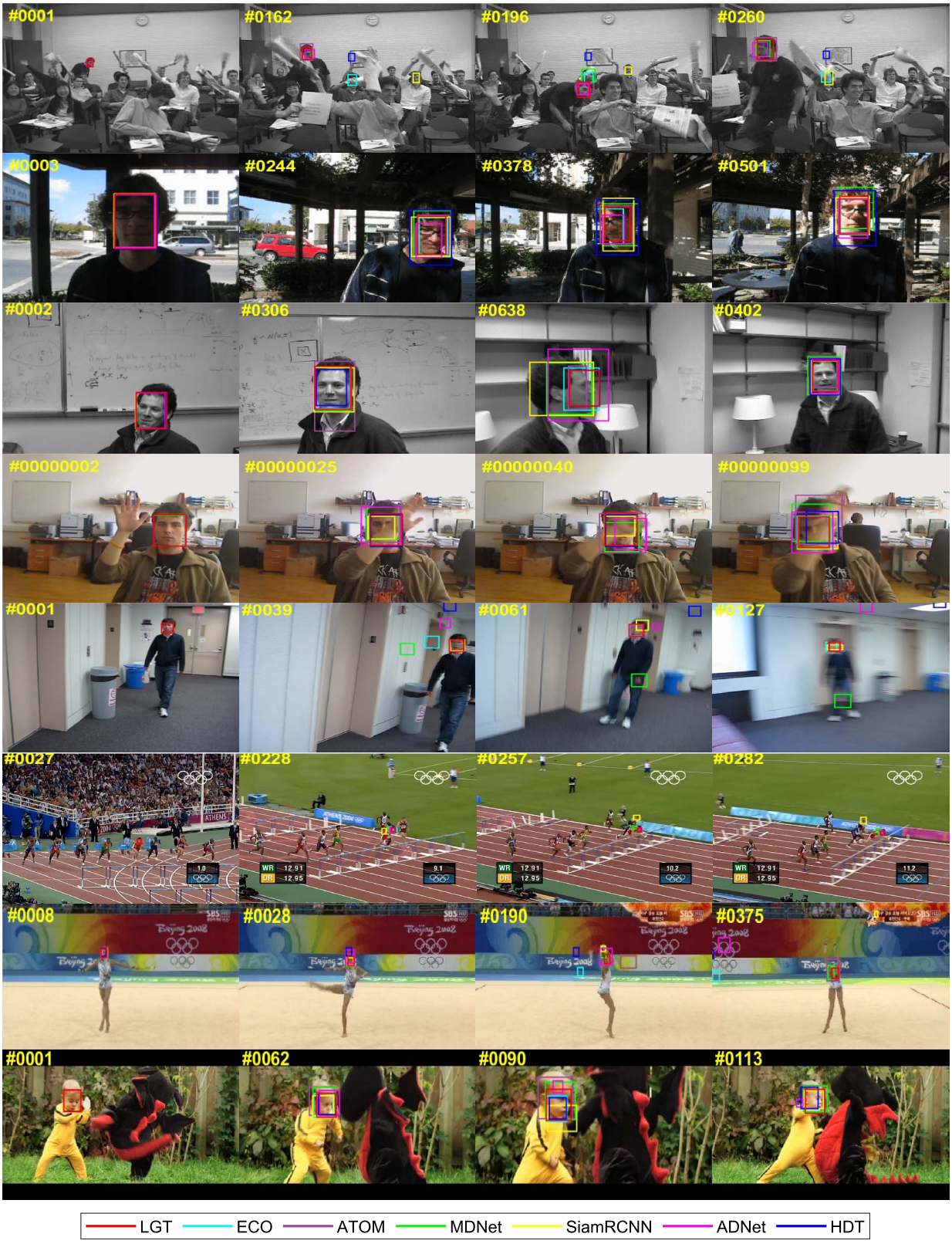

Fig. 8. Sample tracking results on several challenging image sequences (*Freeman4, Trellis, Fleetface, Hand, Blurbody, Hurdle_ce1, Gym,* and *Dragonbaby*).

tracking methods: ATOM in terms of precision, SiamRCNN and ECO in terms of AUC. This indicates the robustness of the proposed method. In particular, LGT outperforms the runner-up, ATOM, by about 3% according to the precision plots (about 8% according to success plots). In light of success plots, LGT also leads the runner-up, SiamRCNN and ECO, about

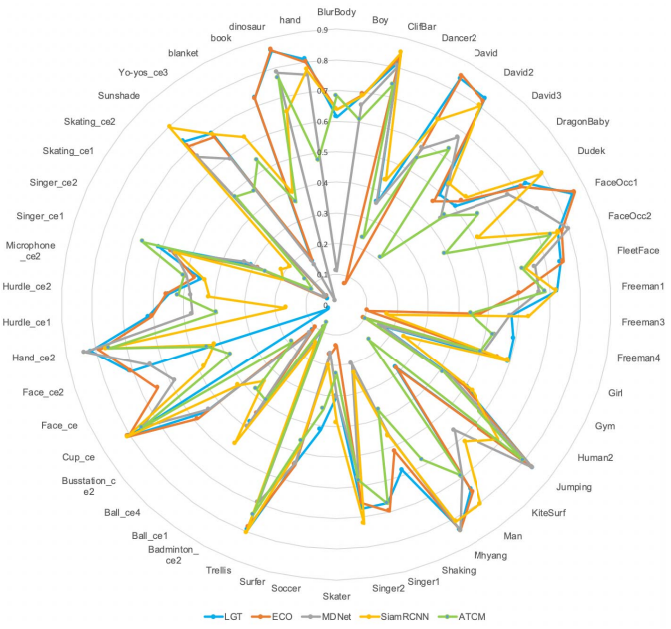

Fig. 9. Tracking results using the average overlap metric on each sequence.

2% (about 6.4% according to precision plots). These results demonstrate the effectiveness and superiority of the proposed method. Considering ECO utilizes the same correlation filter techniques but different CNN features, the success of LGT is attributed to the complementary G-CNN and L-CNNs.

2) Qualitative Evaluation: Several sample tracking results are presented in Figure 8. As it can be seen in the first row, our tracker is able to successfully track the face even when the face occupies a rather small region. The face in the second row undergoes large illumination changes. All the compared methods successfully track the face but our tracker achieves the most accurate tracking bounding box. The faces in the third and fourth rows suffer from pose changes and partial occlusions. The proposed LGT method performs robustly in these challenges. In the fifth and the last rows, the target person moves very fast, which results in blured faces. In such challenging scenarios the proposed LGT method still successfully locates the face while other methods drift to the background. In the sixth and seventh rows, even though the face is too small to be seen clearly, the proposed LGT method can accurately track the face in most frames than other trackers. As discussed earlier, both the L-CNNs and the G-CNNs contribute to the robust performance of the proposed tracking method.

E. Failure Cases and Challenges

As discussed before, the best score in terms of AUC in the success plots is only 55.9%, which is far from satisfaction. To find out the main reasons, we compute the average overlap rate of top-5 trackers on each image sequence and present the results in Figure 9, which shows that there exists eight image sequences that heavily challenge existing trackers. We detail the performance of the evaluated trackers on these eight images sequences in Table VI in term of both the average overlap rate and the average distance error metrics. Table VI shows that the best three trackers perform badly with the overlap rate less than 0.1 on *Badmintion_ce2, Skating_ce1* and *Skating_ce2*, less than 0.2 on *Gym, Ball_ce4* and *Blanket*, around 0.3 on *Skater* and *Singer_ce2*. The bad performance leads to the overall undesirable results on the whole dataset.

To figure out the reasons for tracking failures, we locate the starting frames of heavy drifts on each of these eight challenging image sequences and show the results in Figure 10.

TABLE VI
TRACKING RESULTS ON TOP-8 CHALLENGING IMAGE SEQUENCES

	Average Overlap Rate			Average Error (pixel)		
	LGT	SiamRCNN	ECO	LGT	SiamRCNN	ECO
Gym	0.165	0.244	0.099	10.548	13.621	102.660
Skater	0.301	0.387	0.135	5.618	3.957	55.323
Badminton_ce2	0.084	0.139	0.084	54.719	326.024	54.579
Ball_ce4	0.113	0.345	0.101	90.671	20.693	41.087
Singer_ce2	0.308	0.195	0.282	8.876	70.635	7.380
Skating_ce1	0.035	0.214	0.045	256.963	54.700	152.289
Skating_ce2	0.042	0.195	0.042	134.550	110.398	179.389
Blanket	0.162	0.624	0.167	36.719	2.515	34.136

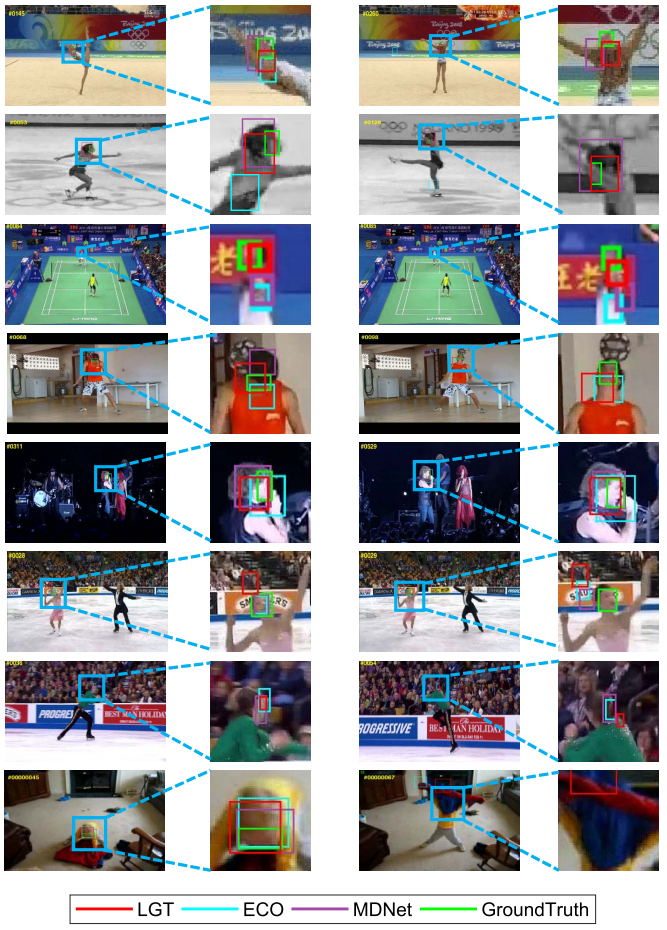

Fig. 10. Critical tracking failures on the top-8 challenging image sequences (from top to bottom: *Gym, Skater, Badminton_ce2, Ball_ce4, Singer_ce2, Skating_ce1, Skating_ce2,* and *Blanket*).

It turns out that heavy occlusion and tiny visible face region lead to these tracking failures on image sequences *Gym, Skater, Ball_ce4*, and *Blanker*. Undiscriminative appearance due to limited resolution (around 16×12 pixels) hampers the tracking methods on image sequence *Badminton_ce2*. As for image sequence *Skating_ce1*, the large translation space (compared to the size of the target face) results in the tracking failure. At the last, similar distractor faces and the long-term un-visibility of the target face (about 20 successive frames) lead to the failure of all trackers on image sequence *Skating_ce2*.

Overall, interfering factors, such as low resolution, tiny visible region, fast motion and occlusion, greatly challenge state-of-the-art methods for face tracking. To address these problems, either a combination of general object tracking and face-specific tracking or the integration of face detection into tracking algorithms may help re-localize the target faces, just like the relatively good results of the detection based tracking method SiamRCNN shown in Table VI.

V. Conclusion

To overcome challenging issues in face tracking, we propose a CNN-based face tracking method in this article, which takes advantage of powerful representations of hierarchical CNN features in both local and global levels. The proposed L-CNNs are designed to extract local features from face regions, such as nose, mouth and eyes. The G-CNN is developed to capture global features from the whole face. The resulting global and local representations within the correlation filter tracking framework are complementary to each other to better distinguish the target face from its background and distractor faces. In addition, to the best of our knowledge, we collect and annotate the largest face tracking dataset, which is far from saturation and facilitates further exploration in this field. Extensive experimental results on the collected dataset demonstrate the effectiveness of the proposed method in comparison to several state-of-the-art trackers.

References

[1] 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.

[2] M. Kristan *et al.*, "The visual object tracking VOT2017 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1949–1972.

[3] 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. 4586–4595.

[4] Y. Yuan, L. Cheng, Z. Wang, and C. Sun, "Position tracking and attitude control for quadrotors via active disturbance rejection control method," *Sci. China Inf. Sci.*, vol. 62, no. 1, pp. 10201:1–10201:10, Jan. 2019.

[5] B. Zhao, Y. Peng, Y. Song, and R. Qin, "Sliding mode control for consensus tracking of second-order nonlinear multi-agent systems driven by Brownian motion," *Sci. China Inf. Sci.*, vol. 61, no. 7, pp. 70216:1–70216:8, Jul. 2018.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[7] G. G. Chrysos, E. Antonakos, P. Snape, A. Asthana, and S. Zafeiriou, "A comprehensive performance evaluation of deformable face tracking 'in-the-wild,"' *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 198–232, Apr. 2018.

[8] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Visual tracking and recognition using probabilistic appearance manifolds," *Comput. Vis. Image Understand.*, vol. 99, no. 3, pp. 303–331, Sep. 2005.

[9] K.-C. Lee and D. Kriegman, "Online learning of probabilistic appearance manifolds for video-based recognition and tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 852–859.

[10] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 1003–1011.

[11] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph*, vol. 33, no. 4, pp. 43:1–43:10, 2014.

[12] G. G. Chrysos, E. Antonakos, S. Zafeiriou, and P. Snape, "Offline deformable face tracking in arbitrary videos," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 954–962.

[13] E. Sánchez-Lozano, B. Martínez, G. Tzimiropoulos, and M. F. Valstar, "Cascaded continuous regression for real-time incremental face tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 645–661.

[14] N.-T. Do, S.-H. Kim, H.-J. Yang, G.-S. Lee, and I.-S. Na, "Face tracking with convolutional neural network heat-map," in *Proc. 2nd Int. Conf. Mach. Learn. Soft Comput. (ICMLSC)*, 2018, pp. 29–33.

[15] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: Tracking-learning-detection applied to faces," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 3789–3792.

[16] S. Lucey, Y. Wang, J. Saragih, and J. F. Cohn, "Non-rigid face tracking with enforced convexity and local appearance consistency constraint," *Image Vis. Comput.*, vol. 28, no. 5, pp. 781–789, May 2010.

[17] P. Vadakkepat, P. Lim, L. C. De Silva, L. Jing, and L. Li Ling, "Multimodal approach to human-face detection and tracking," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1385–1393, Mar. 2008.

[18] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3460–3469.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[20] S. Ren, K. He, R. B. 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.

[21] S. Jin, H. Su, C. Stauffer, and E. Learned-Miller, "End-to-end face detection and cast grouping in movies using Erdös-Rényi clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5286–5295.

[22] W. Kienzle, G. H. Bakir, M. O. Franz, and B. Schölkopf, "Face detection–efficient and rank deficient," in *Proc. NeurIPS*, 2004, pp. 673–680.

[23] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5325–5334.

[24] S. C. Brubaker, J. Wu, J. Sun, M. D. Mullin, and J. M. Rehg, "On the design of cascades of boosted ensembles for face detection," *Int. J. Comput. Vis.*, vol. 77, pp. 65–86, Sep. 2008.

[25] S. Z. Li and Z. Zhang, "Floatboost learning and statistical face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1112–1123, Sep. 2004.

[26] S. Z. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang, "Kernel machine based learning for multi-view face detection and pose estimation," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2001, pp. 674–679.

[27] S. Liao, A. K. Jain, and S. Z. Li, "A fast and accurate unconstrained face detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 211–223, Feb. 2016.

[28] H. Sahbi, D. Geman, and N. Boujemaa, "Face detection using coarse-to-fine support vector classifiers," in *Proc. Int. Conf. Image Process.*, Sep. 2002, pp. 925–928.

[29] X. Zhang, Y. Yang, Z. Han, H. Wang, and C. Gao, "Object class detection: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 10:1–10:53, 2013.

[30] M. Jones and P. Viola, "Fast multi-view face detection," Mitsubishi Electr. Res. Lab, Cambridge, MA, USA, Tech. Rep. TR-20003-96, 2003, p. 2.

[31] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool, "Towards multi-view object class detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 1589–1596.

[32] H. Sahbi and D. Geman, "A hierarchy of support vector machines for pattern detection," *J. Mach. Learn. Res.*, vol. 7, pp. 2087–2123, Oct. 2006.

[33] Y. Amit and D. Geman, "A computational model for visual selection," *Neural Comput.*, vol. 11, no. 7, pp. 1691–1715, Oct. 1999.

[34] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.

[35] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 964–975, Feb. 2018.

[36] A. Jourabloo and X. Liu, "Large-pose face alignment via CNN-based dense 3D model fitting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4188–4196.

[37] Z. Ren, S. Yang, F. Zou, F. Yang, C. Luan, and K. Li, "A face tracking framework based on convolutional neural networks and Kalman filter," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2017, pp. 410–413.

[38] I.-H. Choi and Y.-G. Kim, "Deep manifold embedding active shape model for pose invarient face tracking," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 578–581.

[39] 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. 6931–6939.

[40] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4303–4311.

[41] Y. Qi *et al.*, "Hedging deep features for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1116–1130, May 2019.

[42] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[43] S. McKenna and S. Gong, "Tracking faces," in *Proc. 2nd Int. Conf. Autom. Face Gesture Recognit.*, Oct. 1996, pp. 271–276.

[44] K. Schwerdt and J. L. Crowley, "Robust face tracking using color," in *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Mar. 2000, pp. 90–95.

[45] H. Lee, D. Kim, and S. Lee, "Robust face-tracking using skin color and facial shape," in *Audio- and Video-Based Biometric Person Authentication*. Berlin, Germany: Springer-Verlag, 2003, pp. 302–309.

[46] F. Yang and M. Paindavoine, "Implementation of an rbf neural network on embedded systems: Real-time face tracking and identity verification," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1162–1175, Sep. 2003.

[47] A. Ranftl, F. Alonso-Fernandez, and S. M. Karlsson, "Face tracking using optical flow development of a real-time adaboost cascade face tracker," in *Proc. Int. Conf. Biometrics Special Interest Group*, 2015, pp. 39–48.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[49] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

[50] 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.

[51] N. Wang and D. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 809–817.

[52] 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.

[53] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1349–1358.

[54] 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.*, 2016, pp. 472–488.

[55] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5000–5008.

[56] 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.

[57] 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.

[58] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[60] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[61] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3676–3684.

[62] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.

[63] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–12.

[64] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.

[65] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

[66] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[67] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2015, pp. 41.1–41.12.

[68] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[69] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.

[70] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[71] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[72] M. Kristan *et al.*, "The visual object tracking VOT2013 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 98–111.

[73] M. Kristan *et al.*, "The visual object tracking VOT2014 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2013, pp. 191–217.

[74] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 564–586.

[75] 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.

[76] H. Fan *et al.*, "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5374–5383.

[77] M. Müller, A. Bibi, S. Giancola, S. Al-Subaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 310–327.

[78] N. Xu *et al.*, "Youtube-VOS: A large-scale video object segmentation benchmark," *CoRR*, vol. abs/1809.03327, pp. 603–619, Sep. 2018.

[79] 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.*, early access, Dec. 4, 2019, doi: 10.1109/TPAMI.2019.2957464.

[80] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7464–7473.

Yuankai Qi received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, China, in 2013 and 2018, respectively. His research interests include object tracking, video segmentation, sparse coding, and machine learning. He serves as a Reviewer of top tier conferences and journals, such as the CVPR, NeurIPS, ICCV, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the IEEE TRANSACTIONS ON MULTIMEDIA.

Shengping Zhang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai. He had been a Postdoctoral Research Associate with Brown University and Hong Kong Baptist University, and a Visiting Student Researcher with the University of California at Berkeley. He has authored or coauthored over 60 research publications in refereed journals and conferences. His research interests include deep learning and its applications in computer vision.

Feng Jiang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2001, 2003, and 2008, respectively. He is currently an Associate Professor with the Department of Computer Science, Harbin Institute of Technology, and a Visiting Scholar with the School of Electrical Engineering, Princeton University. His research interests include computer vision, pattern recognition, and image and video processing.

Huiyu Zhou received the B.Eng. degree in radio technology from the Huazhong University of Science and Technology, China, the M.Sc. degree in biomedical engineering from the University of Dundee, U.K., and the Ph.D. degree in computer vision from Heriot-Watt University, Edinburgh, U.K. He is currently a Professor with the School of Informatics, University of Leicester, U.K. He has authored or coauthored over 280 peer reviewed articles in the field. His research work has been or is being supported by U.K. EPSRC, MRC, EU, Royal Society, Leverhulme Trust, Puffin Trust, Invest NI, and industry.

Dacheng Tao (Fellow, IEEE) is currently a Professor of computer science and an ARC Laureate Fellow with the School of Computer Science and the Faculty of Engineering, The University of Sydney. His research results in artificial intelligence have expounded in one monograph and more than 200 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IJCV, JMLR, AIJ, AAAI, IJCAI, NeurIPS, ICML, CVPR, ICCV, ECCV, ICDM, and KDD, with several best paper awards. He is a fellow of AAAS, ACM, and the Australian Academy of Science. He received the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Museum Scopus-Eureka Prize.

Xuelong Li (Fellow, IEEE) is currently a Full Professor with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China.