Noname manuscript No.
(will be inserted by the editor)

Mobile Face Tracking: A Survey and Benchmark

Yiming Lin · **Jie Shen** · **Shiyang Cheng** · **Maja Pantic**

Received: date / Accepted: date

Abstract With the rapid development of smartphones, facial analysis has been playing an increasingly important role in a multitude of mobile applications. In most scenarios, face tracking serves as a crucial first step because more often than not, a mobile application would only need to focus on analysing a specific face in a complex setting. Albeit inheriting many commons traits of the generic visual tracking problem, face tracking in mobile scenarios is characterised by a unique set of challenges, including rapid illumination change, significant scale variation, in-plane and out-of-plane rotation, cluttered background, occlusion, and temporary disappearance of the tracking target. Despite of its importance, mobile face tracking has received little attention in the past. This is largely due to the fact that there has been no suitable benchmark for the development and evaluation of mobile face trackers. In this work, we propose iBUG MobiFace benchmark, the first mobile face tracking benchmark consisting of 50 sequences captured by smartphone users in unconstrained environments. The sequences contain a total of 50,736 frames with 46 distinct identities to be tracked. The tracking target in each sequence is selected with varying difficulties in mobile scenarios. In addition to frame by frame bounding box, the annotations of 9 sequence attributes(e.g. multiple faces) are provided. We further provide a survey of 23 state-of-the-art visual tracker and a comprehensive quantitative evaluation of these methods on the proposed benchmark. In particular, trackers from two most popular frameworks, namely, correlation filter-based tracking and deep learning-based tracking, are studied. Our experiment shows that (a) the performance of all existing generic object trackers drops significantly on the mobile face tracking scenario, suggesting the need of more research effort into mobile face tracking, and (b) the effective combination of deep learning tracking and face-related algorithms(*e.g.* face detection) provides the most promising basis for future developments in the field. The database, annotations and evaluation protocol / code will be made publicly available on the iBUG website[1].

Keywords Mobile face tracking · Survey · Benchmark · Correlation filter tracking · Deep learning tracking

1 Introduction

Face tracking is the process of locating a target face in a video over time. With the rapid development of high-end smartphones, face tracking has served as a crucial first step in mobile applications that apply face analysis, *e.g.* active authentication[51], facial expression analysis, and mobile interaction[23]. Given an initial location of the target face, face tracking aims at estimating the target's unknown states, *e.g.* position and scale, in the subsequent frames. Despite recent successes[44,43], tracking a face in mobile scenarios remains extremely challenging due to the large appearance variations caused by illumination changes, scale variations, in-plane or out-of-plane rotations, heavy occlusions and target disappearance from the camera view, and etc. While some mobile detection benchmarks[51] and generic object tracking benchmarks have been proposed[75], there are, to our best knowledge, no public benchmarks for mobile face tracking. The lack of suitable databases makes it extremely difficult to evaluate a tracker's ability to track a face in mobile scenarios.

Albeit inheriting many common traits from generic object tracking[75], mobile face tracking differ from object tracking in many aspects:

Yiming Lin E-mail: yiming.lin15@imperial.ac.uk
Jie Shen E-mail: jie.shen07@imperial.ac.uk
Shiyang Cheng E-mail: shiyang.cheng11@imperial.ac.uk
Maja Pantic E-mail: maja.pantic@gmail.com
Imperial College London, UK, SW7 2AZ

[1] `https://ibug.doc.ic.ac.uk/people/ylin`

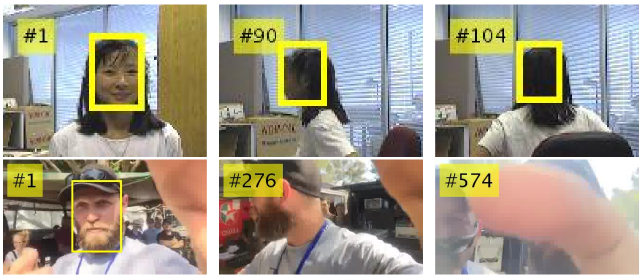

Fig. 1: Annotation difference. Top row: the Girl sequence from a general object tracking database[76] Although the tracking target is a face, when the face rotates out-of-plane, the annotation still gives the hair part. Bottom row: a sequence from our proposed database, we annotate severe out-of-plane rotations and occlusions as [0,0,0,0] since the face has disappeared.

- Due to frequent out-of-plane rotations and / or camera motion, the target face can undergo large scale variations during the tracking session, whereas the target's size and the aspect ratio usually do not change significantly in object tracking benchmarks.
- Objects normally move slowly and the camera usually stands still in object tracking benchmarks[75], whereas the use of hand-held smartphones can result in extremely fast camera motion.
- In object tracking benchmarks, it is relatively rare to have similar objects in the view, whereas in mobile face tracking, the tracker should be able to distinguish between the target face and other faces in the view (see Fig.2).
- In object tracking annotations, the target is always annotated with an approximate bounding box even in severe occlusions or rotations. In contrast, faces with severe occlusions or out-of-plane rotations should not be tracked in face tracking as they make no contribution to the face analysis process (See Fig.1).
- The mobile camera has a smaller field of view compared to conventional digital cameras, thus the face can easily move out of view, *i.e.*, some part or the whole of the target face leaves the view.
- Limited computation capability on the mobile devices requires the tracker to be as efficient as possible.
- Mobile cameras suffer from rolling-shutter effects[22], this means in fast camera motion, there will be more unwanted distortions in addition to motion blur.

A good mobile face trackers should not only be able to tackle conventional challenges (*e.g*. illumination changes) but also able to handle new challenges in mobile scenarios (*e.g*. out-of-view) efficiently.

In this paper, we introduce iBUG MobiFace benchmark, the first mobile face tracking benchmark, which consists of 50 video sequences captured by smartphone users around the world in fully unconstrained environments. There are 50,736 frames from 46 distinct identities to be tracked. The target face is carefully selected with varying difficulties in mobile scenarios. We have manually annotated frame by frame bounding boxes and 9 sequence attributes (*e.g*. multiple faces) for every sequence. State-of-the-art visual trackers which have achieved top performance on recent visual tracking benchmarks (*e.g*. Online Tracking Benchmark (OTB)[75] and Visual Online Tracking (VOT) challenges[35]) are surveyed. These trackers are divided into two categories: (a) Correlation Filter(CF)-based, and (b) Deep Learning(DL)-based. We evaluate 23 representative trackers using the open-source codes from the authors. We provide comprehensive analysis of the results on iBUG MobiFace benchmark. The database, annotations and evaluation protocol / code will be made publicly available on the iBUG website.

Our contributions can be summarised as follows:

- We introduce the iBUG MobiFace benchmark, which is the first of its kind in mobile face tracking. It consists of 50 video sequences and 50,736 frames with 46 identities to be tracked in fully unconstrained mobile scenarios. Evaluation protocols and tools are provided.
- We survey state-of-the-art visual trackers from two popular tracking frameworks, namely correlation filters-based tracking and deep learning-based tracking.
- We conduct comprehensive quantitative evaluations of 23 advanced trackers on iBUG MobiFace benchmark.

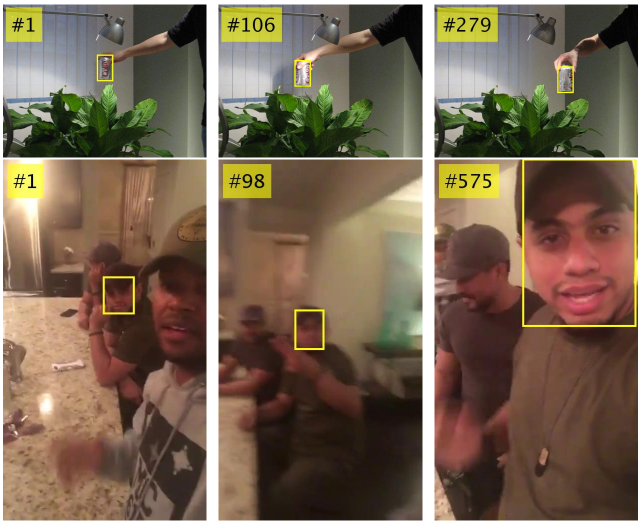

Fig. 2: Differences between general object tracking and mobile face tracking. Top: the Coke sequence from [75]. The camera is set still. The object is moving slowly and the scale remains the same. There are no similar objects. Bottom: one example sequence from our proposed database. The face undergoes severe motion blur and large scale variations due to camera movement. Multiple faces are in the view.

The results indicate that mobile face tracking is far from being solved and the effective combination of deep learning and face-related algorithms can be a potential research direction.

The remaining of this paper is organised as follows. We formulate the mobile face tracking problem in Section 2. Section 3 surveys the visual trackers based on (a) correlation filters and (b) deep learning, which have become the state-of-the-art tracking frameworks in the past decade. In Section 4, we introduce the iBUG MobiFace dataset and evaluation protocols. In Section 5, we evaluate 23 advanced trackers on the proposed benchmark and discuss the results. Finally, Section 6 concludes the the survey and benchmark and discusses future directions for mobile face tracking.

2 Problem formulation

We formulate the mobile face tracking problem using the tracking-by-detection paradigm[35,75,76]. The bounding box of the target face is given in the first frame and the goal of mobile face tracking is to find the optimal location and size of the target face in the t-th frame ($t > 1$), represented as a rectangle r_t, that achieve the highest score within the candidate rectangle set $\mathscr{R}_t$:

$$r_t = \begin{cases} \emptyset, if\ conditions \\ \arg\max_{r \in \mathscr{R}_t} \{Score(f(x_t, r)|\theta_{t-1})\}, otherwise \end{cases} \tag{1}$$

The *conditions* include the cases where the target face are not observable because of temporary disappearance (becoming out-of-view), out-of-plane rotations and / or severe occlusions. The function *Score* gives the score of the candidate region r of frame x_t given the model parameters from the last frame θ_{t-1}, while f is a certain image transformation function that transforms the rectangle r. We assume that the location of first frame x_1 is provided in the tracker initialisation stage. Once the position of the target is located, the model parameters is usually updated by minimising a loss function $\mathscr{L}(\theta; \mathscr{X}_t)$:

$$\theta_t = \arg\min_{\theta \in \Theta} \{\mathscr{L}(\theta; \mathscr{X}_t) + \lambda R(\theta))\} \tag{2}$$

where $\mathscr{X}_t = (x_i, r_i)_{i=1}^t$ is the set of historical frames and corresponding rectangles. When $i = 1$, $\mathscr{X}_1$ represents the groundtruth. $R(\theta)$ with its relative weight λ is used to regularise the model parameters so as to prevent over-fitting. To locate the target accurately and efficiently, the functions in (1) and (2) have to be carefully designed.

3 Visual trackers survey

In this section we survey state-of-the-art visual trackers from two most popular frameworks, the correlation filter-based approaches and deep learning-based approaches. Trackers from these two frameworks have achieved top performance on the OTB and VOT visual tracking benchmarks[76,35]. Since face is one of the commonly seen objects in those benchmarks, the generic object trackers are expected to provide a solid base to support further investigation into face tracking. The face tracking problem has received less attention compared to face detection[32] and face recognition[59]. In the literature, only few face trackers are developed [43, 44,73]. This is possibly due to a common belief that the face tracking problem could be solved by combining a face detector with a face recognition system. However, the efficacy of this naive approach remains doubtful, as it treats each frame as independent and identically distributed and ignores the spatio-temporal information in the video. We argue that, to efficiently track the face, face trackers should based on the successful trackers developed by the object tracking community.

3.1 Correlation filter-based tracking

Discriminative Correlation Filters (CFs) have been widely used in many computer vision problems [37]. Recently, CF-based trackers has achieved outstanding performance on both the VOT challenges[35] and the OTB benchmark[76]. A CF can be considered as a holistic appearance encoder and the tracking process is a template matching process. In the initialisation, the CF is trained with the patch cropped from the target in the first frame. A desired output is used as the optimisation target. Next, the CF is convolved with a candidate window that cropped based on the previous location in the subsequent frames. A spatial confidence response map can be generated and the location of the highest value is considered as the prediction in this frame. The CF is then updated according to the prediction.

We denote the features of the training patch as $f(\mathbf{x}) \in \mathbb{R}^{M \times N \times D}$ and the desired output $\mathbf{y} \in \mathbb{R}^{M \times N}$ that is a Gaussian-shaped function peaked at the target centre. For initialisation, the optimal CF $\mathbf{w}^*$ is obtained by minimising the ridge regression loss:

$$E = \left\| \sum_{d=1}^{D} \mathbf{w}^d \circledast f^d(\mathbf{x}) - \mathbf{y} \right\|^2 + \lambda \sum_{d=1}^{D} \left\| \mathbf{w}^d \right\| \tag{3}$$

where $\mathbf{w}^d$ is the d-th channel of the CF, $\circledast$ refers to circulant convolution and λ is the constant regularisation factor. Based on the Convolution Theorem and circulant data assumption[26], the solution can be computed as Hadamard product in the Fourier domain[10]:

$$\hat{\mathbf{w}}^d = \frac{\hat{f}^d(\mathbf{x}) \odot \hat{\mathbf{y}}^*}{\sum_{d=1}^{D} \hat{f}^d(\mathbf{x}) \odot (\hat{f}^d(\mathbf{x}))^* + \lambda} \tag{4}$$

Tracker	Improvements	FPS	GPU	Implementation	Link
BACF[18]	B	40	No	M	http://www.hamedkiani.com/bacf.html
CACF[53]	B	35	No	M	https://ivul.kaust.edu.sa/Pages/pub-ca-cf-tracking.aspx
CCOT[15]	F + S	<1	Yes	M + m	https://www.cvl.isy.liu.se/research/objrec/visualtracking/regvistrack/
DSST[10]	S	24*	No	M	http://www.cvl.isy.liu.se/research/objrec/visualtracking/scalvistrack/index.html
ECO[9]	F + S	8	Yes	M + m	http://www.cvl.isy.liu.se/research/objrec/visualtracking/ecotrack/index.html
ECO-HC[9]	F + S	8	No	M	http://www.cvl.isy.liu.se/research/objrec/visualtracking/ecotrack/index.html
KCF[26]	F	320	No	M	http://www.robots.ox.ac.uk/∼joao/circulant/
HCFT[47]	F	11	Yes	M + m	https://github.com/jbhuang0604/CF2
HDT[55]	F	No	Yes	M + m	https://sites.google.com/site/yuankiqi/hdt/
IBCCF[38]	B + S	1	Yes	M + m	https://github.com/lifeng9472/IBCCF
LCT[50]	L	27	No	M	https://github.com/chaoma99/lct-tracker
MUSTer[29]	L	4	No	M	https://sites.google.com/site/multistoretrackermuster/
RPT[41]	S	4	No	M	https://github.com/ihpdep/rpt
SAMF[40]	S	7	No	M	https://github.com/ihpdep/samf
SRDCF[11]	B	<1	No	M	http://www.cvl.isy.liu.se/research/objrec/visualtracking/regvistrack/index.html
Staple[2]	B	80	No	M	https://github.com/bertinetto/staple

Table 1: The table reports the short name of the benchmarked tracker, its improvements, whether it uses GPU, the frame per second (FPS) from the original paper, the implementation as well as the link to the implementation. The initials stand for: P-Pioneering, F-Feature improvement, S-Scale handling, B-Boundary effect handling, L - Long-term components, M - MATLAB, m-matconvnet[68]. The star sign after the FPS stands for the median speed. These trackers are evaluated on the mobile face tracking database in Section 5

where $\hat{\mathbf{y}}$ refers to the Fourier transform of $\mathbf{y}$, $\odot$ denotes Hadamard product and $*$ denotes the complex conjugate.

In the detection process, a search window $\mathbf{z}$ is cropped and the features $f(\mathbf{z})$ are used to compute the response map:

$$\mathbf{g} = \mathscr{F}^{-1}\left(\sum_{d=1}^{D} \hat{\mathbf{w}}_{t-1}^{*d} \odot \hat{f}^{d}(\mathbf{z})\right) \tag{5}$$

where $\hat{\mathbf{w}}_{t-1}$ is the CF trained from frame 1 to frame $t-1$ and $\mathscr{F}^{-1}$ denotes the inverse FFT. The position of the maximum value of $\mathbf{g}$ is the prediction in this frame. From this prediction $\mathbf{w}_{t-1}$ is updated using different update schemes, *e.g.* running average, to obtain $\mathbf{w}_t$.

In the following section, we first review the pioneering works that integrated CFs into tracking problem. After that, based on the major contributions to the CF tracking framework, we review the trackers from five perspectives: (a) **Feature**: different features exploited; (b) **Scale handling**: the strategy used to handle scale variations; (c) **Boundary effects handling**: the strategy used to handle boundary effects; (d) **Long-term components**: the additional component used to allow for long-term tracking.

3.1.1 Pioneering works

MOSSE CFs were first successfully applied to object tracking in [4] where a CF is trained in the frequency domain by minimising the sum of the squared error of the CF output and the desired output. The training examples are generated by randomly rotate the predicted region (using affine transform) in the previous frame, and the desired output of each example was given by a Gaussian-shape function with the peak at the centre of the target. The CF has a closed-form solution in the frequency domain. The learned weights of the CF were then updated by running average in every subsequent frame. In the frequency domain, the correlation between the tracking window and the CF can be computed by element-wise product, allowing the MOSSE tracker to run at extreme speed (669 FPS) [4] while achieving competitive tracking results.

CSK Even though MOSSE is extremely fast and efficient, the CF in MOSSE is barely a simple linear classifier, which is not discriminative enough to serve as the kernel-based classifier, thus lowering the robustness of the trackers. The kernalised CFs were introduced in the CSK tracker [25]. Efficient dense sampling of training data is done by exploiting the circulant matrix of the target region. One main advantage of circulant matrices is that they can be diagonalised in the Fourier domain using the DFT matrix and the base vector. The optimisation of a CF is reformulated as a kernel ridge regression formulation problem. The solution is closed-form as long as the kernel matrix is also circulant. The idea of modelling the movement of the target as circulant matrix takes numerous commonly encountered negative examples into account, while the kernelised formulation allows the pixel intensity of training data to be projected to a higher-dimensional kernel space. Hence the learned CF is able to better distinguish the target from the background.

3.1.2 Features

In MOSSE and CSK, raw pixel intensities are used to train the CF, which are usually very sensitive to noises. Various features have been exploited to improve the CF-based trackers.

Hand-crafted features The multi-channel HOG features[8] were first applied in the KCF [26] tracker to replace the pixel intensities as input in the CSK tracker. The CN [14] tracker also extended the CSK tracker using 11-dimension linguistic colour features, *i.e*., the colour names[74] as input. After that, the CN and HOG features were combined in the SAMF[40] tracker which won the second place in the VOT2014 challenge[36]. Such HOG and CN feature combination has turned out very effective for tracking and is also applied in [65]. Bertinetto *et al.* [2] proposed that template features and colour statistics can be complementary, based on the observation that colour statistics are more robust to fast motion but not discriminative enough when the colour distribution of the background is similar to that of the target. On the other hand, template-based features (e.g. HOG) can discriminate the object from the background with similar colour distributions but perform poorly when the target moves rapidly. The resulted tracker, called staple, combines linearly HOG features with a global colour histogram as the input features to the CF. Experiments showed that Staple outperformed most CF-based trackers which use hand-engineered features on multiple benchmarks [2,35,76].

Convolutional features Convolutional features extracted by Convolutional Neural Networks(CNNs) have proven very robust in computer vision tasks, including image recognition [61,24] and object detection[20,58]. However, a large amount of training data with annotations is required for deep CNNs to learn image representations so as to avoid overfitting. This is normally not the case in the online tracking problem where the first frame is the only training instance available. To exploit the capability of feature extraction of CNNs, many trackers have utilised the CNNs pre-trained on large datasets, *e.g*. VGGNet[61], to obtain a discriminative representation of the target.

In [13], Danelljan *et al.* propose the DeepSRDCF tracker based on the SRDCF framework[11] to build correlation filters on the features extracted by the first convolutional layer of the VGGNet[61]. The results of the tracker indicated that the features from the early convolutional layers were more suitable for tracking than features from deeper layers and to HOG features. Ma *et al.* [47] found that while the earlier layers are capable of capturing fine-grained spatial details, the deeper layers can capture more semantics of the target. Hence in [47], the HCFT tracker is proposed that exploits the hierarchical structure of the VGGNet model. Multiple CFs are trained on three different convolutional layers of the VGGNet model(Conv3-4, Conv4-4 and Conv5-4). The response maps of different layers are then collected to conduct a coarse-to-fine search for the maximum response from the deepest layer to the earliest layer, locating the target in the input frame. The hierarchical convolutional features have also been adopted in [55,80,49]. While the convolutional feature maps of different spatial resolutions can be connected by a coarse-to-fine search, the single-resolution input assumption of CFs prohibits the above trackers from fusing the feature maps to produce a joint response. To solve this problem, a multi-resolution feature fusion strategy is proposed in the CCOT tracker [15]. The convolutional feature maps of different resolutions are implicitly interpolated by a learned convolutional operator to have the same spatial resolution, hence the outputs of the CFs on different layers can then be linearly combined into a continuous response map. However, the interpolation process introduces heavy computation burden. A factorised convolution operator is introduced in [9] to speed up the interpolation process. The factorised operator significantly reduces the number of parameters in the CFs, but also allows the effective fusion of convolutional features and hand-engineered features (*e.g*. HOG and CN). Experiments have shown the resulted ECO tracker has outperformed various state-of-the-art trackers on recent benchmarks[9,33]. The simplified version of ECO, called ECO-HC, which uses hand-engineered features is able to run at 60 FPS with better accuracy than Staple[2]. The CFCF tracker, [21], the winner of VOT2017[33], is also based on the factorised convolutional operator.

3.1.3 Scale handling

In conventional CF-based trackers such as MOSSE and KCF, the scale of the target is fixed and the trackers are likely to fail when the target suffer from large scale variations. Numerous trackers have employed different strategies to handle scale variations.

Scale pool A scale pool is proposed in the DSST tracker [10] to handle scale variations. It considers the tracking process as two subproblems, *i.e*., translation estimation and scale estimation. For translation estimation, a multi-dimensional correlation filter is trained on HOG features of the search region, which is similar to KCF[26], to predict the target position. For scale estimation, the search region is first resized into 32 scales defined in the scale pool by bicubic interpolation. At each scale, the HOG features are computed and then reshaped into a one-dimensional vector. Next, a one-dimensional CF is trained on every feature channel and the scale with the maximum response is chosen as the predicted scale. The translation filter and the scale filter are updated separately during tracking. The fDSST tracker[12] further improves the scale searching speed by several approaches such as sub-grid interpolation of correlation scores. Similarly, a scale pool that consists of 7 values ranged from 0.985 to 1.105 is employed in the SAMF tracker[40]. The search region is resized into every scale where different filters are learned. The same idea of the scale pool is also exploited in [65,47,78,11,15].

Part-based tracking Part-based tracking has been employed in many trackers to handle scale variations. In [42], Liu *et al.* propose to divide the target into several parts each of which was tracked by a KCF[26]. These resulting response maps are weighted and integrated into a Bayesian inference framework for tracking. The scale is handled in the target inference process from multiple parts. In the RPT tracker[41], the reliability of each part is modelled by a sequential Monte Carlo framework. Based on the reliable patches, a Hough voting-like scheme is then used to estimate the scale of the target. The similar idea of part-based tracking is also used in [1,45] to handle scale variations.

Despite the above developments in scale handling, an underlying assumption in these trackers is that the aspect ratio of the target remains fixed over time. This is normally true for general objects captured from a long distance. However, for mobile face tracking, the target face can turn sideways frequently, resulting out-of-plane rotations which changes the aspect ratio. Very recently, Huang *et al.* [31] proposed to integrates the class-agnostic detection proposal method called Edge Boxes[82], into the CF tracker to handle the aspect ratio changes.

3.1.4 Boundary effects handling

Unwanted boundary effects are introduced by the underlying periodic assumption of circulant data in the CF-based trackers. When the training data is constructed by the circulant matrix of the target region, the information near the boundary is ignored. This severely reduces the diversity of negative examples when training the CFs, thus leading to over-fitting. As a consequence, if the target moves abruptly to the boundary of the search region, the tracker is likely to drift to distortions that are not seen in the training data. Numerous strategies have been proposed to alleviate the boundary effects. The SRDCF[11] tracker introduces a spatial regularisation methods to penalise the CF weights near the boundary. By the spatial regularisation, a larger set of negative examples can be included in training, thus resulting a more discriminative model. However, since the regularisation formulation breaks the circulant data assumption, the optimisation no longer holds a closed-form solution. The resulting tracker is able to better track the object in fast motion but the tracking speed is severely prohibited by the iterative process. In[19], Galoogahi *et al.* uses the Alternating Direction Method of Multipliers (ADMM) to implicitly zero-pad the single-channel CF. This method is further extended to multi-channel CFs in [18] to take more background information into account. In[46] a spatial reliability map is estimated to constrain the training of correlation filters.[53] proposes a context-aware framework that allows CF trackers to expand the search region without adding computational cost. This is achieved by carefully formulating the contextual information in the filter learning stage with a closed-form solution.

3.1.5 Long-term component

As discussed in Section 1, the target can undergo significant appearance variation due to heavy occlusions, abrupt motion and out-of-view in complicated tracking scenes. Many methods have proposed to use an additional re-detection component to re-detect objects in case of tracking failure. An random fern classifier is introduced in the LCT tracker[50] to complement the CFs by re-detecting the target over the entire image if the maximum response of the CFs is below a threshold. Using the same re-detector, an additional long-term CF is trained in [48] using a conservative learning rate to maintain the target appearance. The response of this long-term CF is then used to trigger the re-detector. A biologically-inspired MUlti-Score Tracker (MUSTer) is proposed in [29] that uses a key point-matching tracker to keep the long-term information. Key points are stored and discarded based on a forgetting curve. When the maximum response of CFs falls below a threshold, the key points are used for matching the target. Zhu *et al.* [81] propose a CUR filter for re-detection which computes the low rank approximation of the large matrix formed by the historical object representations during tracking. While all the above trackers rely on the response of the short-term CFs as the re-detection threshold, Li *et al.* [39] argues the distortions in the response can cause CFs to adapt to the noise with high confidence. This means the long-term component would not be triggered given a fixed threshold. To solve this, a normalised correlation response is proposed in [39] that allows the tracker to better detect distortions in the response map such that the long-term part can be activated promptly.

3.2 Deep learning-based tracking

Deep Learning (DL) has brought significant breakthroughs in many computer vision tasks, including object detection[20, 58] and object recognition[61,24]. Many trackers that apply deep learning have achieved impressive results on VOT and OTB benchmarks[33,76]. This section surveys the best-performing DL-based trackers. We categorise them by their deep learning architectures into four sub-classes: (1) Single CNN-based tracking: trackers that use one single CNN; (2) Siamese CNN-based tracking: trackers that use Siamese CNNs to match the target within the search region; (3) RNN-based tracking: trackers that exploit Recurrent Neural Networks (RNNs) to capture spatio-temporal information; (4) RL-based tracking: trackers that exploit Reinforce Learning

Tracker	Structure	Training	FPS	Implementation	Link
ADNet[77]	RL	Offline+Online	<1	M + m	https://github.com/hellbell/ADNet
CREST[63]	Single CNN	Offline+Online	1	M + m	https://github.com/ybsong00/CREST-Release
CFNet[67]	Siamese CNN	Offline+Online	75	M + m	https://github.com/HyeonseobNam/MDNet
DCFNet[56]	Siamese CNN	Offline+Online	60	M + m	https://github.com/foolwood/DCFNet
DVNet[59]	Detection + Verification	Offline	5	P	https://github.com/davidsandberg/facenet/
MDNet[67]	Single CNN	Offline+Online	<1	M + m	https://github.com/bertinetto/cfnet
SiamFC[3]	Siamese CNN	Offline	58	M + m	https://github.com/bertinetto/siamese-fc

Table 2: The table reports the short name of the evaluated DL-based tracker, its network structure, the network training scheme, the frame per second (FPS) from the original paper, the implementation as well as the link to the implementation. The initials stand for: P-Python, M - MATLAB, m - matconvnet[68].

(RL) to learn the decision-making policy for tracking. We exclude the trackers that use pre-trained CNN as the feature extractor in the CF framework since they have been discussed in Section 3.1.2.

3.2.1 Single CNN-based tracking

These trackers estimate the target states based on the extracted features by a single CNN. The DLT tracker[72], one of the pioneering works of applying deep learning to visual tracking, employs a deep autoencoder to learn compact features of the target. The autoencoder was trained offline on auxiliary data and the trained encoder is used for feature extraction during tracking. A Sigmoid layer is added the encoder to perform binary classification on candidate windows proposed by particle filtering. The Sigmoid layer is updated using stochastic gradient descent(SGD) using positive and negative examples drawn during tracking. Similar training scheme is used in the MDNet tracker[54] where a CNN with shared layers and multiple domain-specific branches was pre-trained on a large tracking database. The domain-specific branches are also fine-tuned on the training examples drawn during tracking. Both the DLT and MDNet tracker rely on numerous particles to cover possible target locations since the spatial information is lost in extracted deep features. A target-specific saliency map is computed in the CNN-SVM tracker[28] to restore the spatial configuration from the input frame. The target is located through sequential Bayesian filtering on the historical saliency maps. Furthermore, the CREST tracker[63] learns spatial residual information by additional residual mapping layers. The whole CNN in CREST is able to be trained end-to-end during tracking because of the residual training strategy[24]. Similar ideas of using one single CNN can also be found in [69,70].

3.2.2 Siamese CNN-based tracking

A Siamese CNN[5] is a Y-shaped CNN that joins the outputs of two identical CNN branches to produces a single output. Siamese CNNs-based trackers[66,67,3,56] have showed excellent performance on VOT and OTB mainly due to their ability to model similarity between objects.

In the SINT tracker[66], two identical query-and-search CNN were proposed with five convolution layers, two max-pooling layers,three region polling layers, one fully-connected layer and a final L^2 layer. The margin contrastive loss was used to train off-line the CNNs on images of objects from ALOV[62] to learn a matching function. During tracking, the CNN weights are frozen and the matching function is used as is. Candidates are sampled at different radial positions and different scales, and are fed to the CNNs to measure the similarity between the query and the target patch in the first frame. Finally, a ridge regressor refines the position and scale of the best-matching candidate based on the maximal inner product to the target patch. The matching function enables SINT to perform long-term tracking in which the target disappears for a while.

In the SiamFC tracker[3], a novel bilinear layer was proposed to compute the cross-correlation of two inputs. The whole Siamese network is made fully convolutional and the output is an correlation response map with high values at pixel locations of the target. The network was trained on a large object detection video dataset[16] and data augmentation considering scale variations was conducted. During tracking, the tracker searches for the object over five scales and updates scales by linear interpolation. The CFNet tracker [67] extended SiamFC by adding a CF and a crop layer to the end of the template branch. To allow for end-to-end training, the CF is formulated as a differentiable layer and the training is done as in SiamFC. While both SINT and SiamFC use the initial appearance of the target, the template in CFNet is updated in each frame by moving average.

The aforementioned Siamese trackers are all use offline pre-trained networks for feature extraction, which utilises prior knowledge independent of the tracking process. The Discriminative Correlation Filters Network (DCFNet) was proposed in [56] that learns convolutional features and performs correlation-based tracking simultaneously. This is achieved by reformulating CF as a special layer added to the Siamese CNN and carefully deriving the backpropagation graph for online training. The correlation filter layer is adaptively updated during tracking. Since the backpropagation is derived in the Fourier domain, the DCFNet tracker can run at 60

	UMDAA-02[51]	OTB50[75]	OTB100[76]	VOT14[36]	VOT15[34]	Ours
Target	Face	Generic	Generic	Generic	Generic	Face
#Sequences	8,756	51	100	25	60	50
Minimum frames	1	69	69	171	48	372
Maximum frames	9	3,870	3,870	1,215	1,506	2,161
Mean frames	4	579	588	414	276	1014
Total frames	33,209	29,490	59,040	10,380	21,870	50,736
Shooting device	Mobile	General	General	General	General	Mobile

Table 3: Comparison between other tracking databases with the proposed mobile face tracking database.

FPS on GPU while achieving competitive performance to SiamFC on VOT and OTB[56].

3.2.3 RNN-based tracking

The recurrent neural network (RNN) [6] is a class of neural network that is suitable for modelling temporal behaviour in sequence data, such as speech and handwriting. There are few works that have attempted to apply RNN to model spatio-temporal information in the visual tracking problem. Multi-directional RNNs were employed in [7] to track the object parts from different directions. The output of multi-dimensional RNNs is joint with the response map of a CF to update the tracker during tracking. The similar RNN structure is used in [17] while the outputs of RNNs are concatenated with those of CNNs to obtain a robust feature representation, which is used to discriminate the object from background in sampled candidate patches. In contrast to modelling the spatial semantics of sequences, ROLO uses LSTM[27], a variant of RNN, to impose temporal constraints on the output of YOLO[57], a efficient object detector, to regress the tracking results.

3.2.4 RL-based tracking

Reinforcement Learning [64] is a set of algorithms that learns a decision-making policy by maximising future rewards. Combined with deep neural networks, RL has been successful in intelligent tasks, such as playing Go[60] or video games[52]. Very recently, some trackers have applied RL and shown promising results. In the ADNet tracker[77], visual tracking is modelled as an action-selecting process, in which the tracker learns a policy that selects optimal actions to track the target from the current state. The network is trained by supervised learning to predict a optimal action given the current state, then by reinforcement learning to adapt itself to action dynamics. During tracking the fully-connected layers are fine-tuned as in MDNet[54] to adapt to appearance changes. Huang *et al.* [30] propose to learn an agent that decides which layers of a deep hierarchical CNN should be used to track the target. Based on the decision, the tracker can use less layers to extract features when the target is easy to track, and more layers when hard. Hence the computation is significantly reduced compared to making a full inference in every frame during tracking.

4 iBUG MobiFace Benchmark

4.1 Overview

The iBUG MobiFace benchmark comprises 50 videos with a total of 50,736 frames. To our knowledge, this is the first mobile face tracking benchmark in the literature. The unique data source guarantees the benchmark to reflect real-world challenges in mobile face tracking. Table 3 compares the proposed dataset with one mobile face detection and 4 generic visual tracking datasets. The average length of video sequences is 1,014, which is significantly more than other datasets. 46 face identities are labelled in the sequences. In addition to frame by frame bounding boxes, 9 sequence attributes are also annotated (See Table 4). The distribution of the attributes is demonstrated in Table 5.

4.2 Dataset

4.2.1 Collection methodology

The videos are collected from mobile live-streaming recordings from YouTube using YouTube Data API [2]. All videos were recorded either in 2017 or 2018 by a variety of contemporary smartphone models. The videos consist of subjects live-streaming under various scenarios and interacting with the audience. The complex scenes are extremely challenging and the motion of the faces is natural and spontaneous. The specific requirement in capturing devices allows the dataset to reflect real-world challenges in mobile face tracking.

4.2.2 Annotation protocol

After downloading the videos, the target face was carefully selected to reflect different levels of difficulties in the mobile scenarios. Given the application of mobile face tracking is to provide consistent bounding box information of the target face for face analysis, the annotation process followed

[2] `https://developers.google.com/youtube/v3/`

Attribute	Description
IV	Illumination Variation - significant illumination change on the target face.
SV	Scale Variation - the area ratio of two bounding boxes in two consecutive frames is smaller than 0.7.
OCC	Occlusion - the face is partially or fully occluded.
FM	Fast Motion - the distance of the target centre is larger than 10% of the frame width between consecutive frames.
IPR	In-Plane Rotation - the face rotates in the image plane.
OPR	Out-of-Plane Rotation - the face rotates out of the image plane.
OV	Out-of-View - part or all of the target leaves the view.
MF	Multiple Faces - other faces exist in the view.
BL	Blur - the target face is blurred due to the motion of target or smartphone, out-of-focus and low resolution.

Table 4: Nine attributes

three broad guidelines: (1) the bounding box is required to tightly contain the forehead, chin, and cheek, while ears are excluded; (2) the location of the faces in which the out-of-view part of the target face is over 90% should be annotated as $[0,0,0,0]$; (3) the location of faces with heavy occlusion or with heavy out-of-plane rotation, in which over 90% of the facial features disappear, should be annotated as $[0,0,0,0]$. A dedicated annotation tool has been developed[3] and each sequence is annotated with rectangular bounding boxes in every frame.

The annotation was performed in a semi-autonomous manner. Specifically, we first ran the ECO tracker[9] on all sequences to get the approximate bounding box of the target faces as the initialisation for manual correction. Two annotators then manually went through all frames to adjust the bounding boxes to the correct location. The final results have been cross-validated by a third annotator. In addition, we annotated each video with 9 visual attributes, which are summarised in Table 4. The distribution of the nine attributes is shown in Table 5.

	IV	SV	OCC	FM	IPR	OPR	OV	MF	BL
IV	**29**	20	20	10	11	27	15	21	19
SV	20	**35**	17	11	13	34	22	26	21
OCC	20	17	**27**	5	10	24	11	22	17
FM	10	11	5	**14**	4	13	10	10	8
IPR	11	13	10	4	**18**	16	12	13	10
OPR	27	34	24	13	16	**47**	26	36	28
OV	15	22	11	10	12	26	**29**	25	21
MF	21	26	22	10	13	36	25	**40**	27
BL	19	21	17	8	10	28	21	27	**30**

Table 5: Distribution of 9 sequence attributes.

4.3 Evaluation protocols

Precision plot, success plot and frame per second (FPS) are used as main metrics to quantitatively evaluate the surveyed tracking methods.

Precision plot. Precision is a widely used evaluation metric on tracking[83]. It is defined as the average Euclidean distance between the centre locations of the tracked face and the groundtruth over all frames of a sequence. The precision plot[75] shows the percentage of frames in which the centre of the estimated location is within the given threshold distance of the centre of groundtruth. Empirically, the tracker's precision at 20 pixel threshold is considered as its representative score. **Success plot.** The success plot [75] shows the percentage of frames in which the intersection of union (IoU) of the predicted and groundtruth bounding boxes is greater than a given threshold. Denoting the groundtruth bounding box as r_{GT} and the predicted bounding box as r_t, the IoU metric is defined as $IoU = \frac{r_{GT} \cap r_t}{r_{GT} \cup r_t}$ where $\cap$ and $\cup$ represent the intersection and union of two regions, respectively. The threshold ranges from 0 to 1. A representative score for each tracker is the area under the curve (AUC) of the success plot. **FPS.** FPS is the average speed of the evaluated tracker over all sequences. The initialisation time in the first frame is not considered and the FPS is computed from the second frame for each sequence. The mobile face trackers should be able to run at high FPS so they can be deployed on mobile devices.

For mobile face tracking, we consider AUC to be a more important creterion than the precision score, because the rich features that can be extracted from the face region (as defined by the bounding box) are usually much more informative to subsequent facial analysis step than what can be captured by face trajectory (as provided by the face's centre point) alone.

5 Experiment

Evaluated trackers 22 state-of-the-art object trackers were evaluated on iBUG MobiFace benchmark. They were selected based on each subcategory discussed in Section 3 that have achieved superior performance on tracking benchmarks[76, 33] . The selected CF-based trackers are listed in Table 1 while the DL-based trackers in Table 2 . For completeness,

[3] `https://github.com/yl1991/bounding_box_tool`

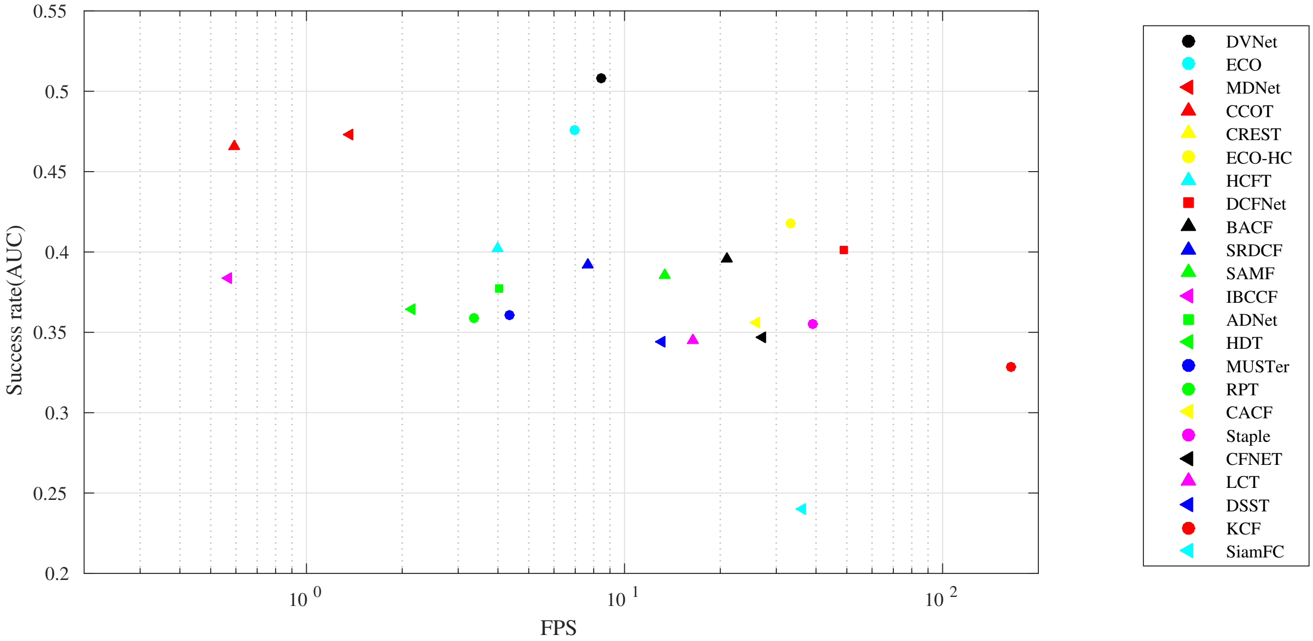

Fig. 3: Speed comparison of different trackers.

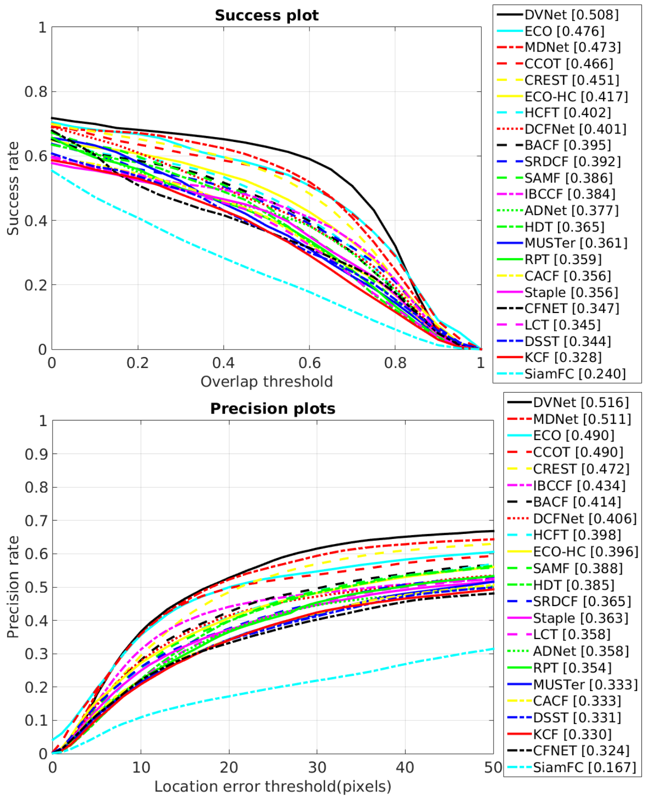

Fig. 4: Overall performance of 23 evaluated trackers on MobiFace with success plot and precision plot. For success plot, the trackers' name is shown with their corresponding AUC. For precision plot, the score at 20 pixel threshold is shown.

we also built a trivial tracker by combining MTCNN[79], a widely-used face detector with Facenet[59], a good-performing face verification algorithm. We call it *DVNet* in the following evaluation process. The default parameters used in the original publication were adopted in our experiment. All trackers were evaluated on an Ubuntu desktop with an Intel(R) Core(TM) i7-7700 3.60GHz CPU and a GeForce GTX 1060 GPU with 3GB memory.

5.1 Overall Performance

The success and precision plots of all trackers are shown in Fig.4. One immediate observation is that no tracker achieved good performance on our benchmark, indicating a need for further studies into this topic. In face, the performance of all evaluated trackers drops significantly on mobile face tracking compared to their performance on the generic object tracking. For instance, the AUC of ECO is 0.69 on OTB100[76] whereas it reduces to 0.47 on our benchmark. SiamFC is ranked the worst in our evaluation in both plots although it is the winner of the VOT17[33] real-time challenge.

Surprisingly, DVNet, the naive combination of face detection and face verification, outperforms all other trackers in terms of both precision and success. This, however, does not mean the trivial approach is the promising direction to go. First of all, in both criteria, the performance of DVNet is hardly satisfactory. Secondly, since face detection algorithms do not take spatio-temporal dynamics into account, a closer inspection shows that DVNet failed in many case when the target could be easily tracked by other visual track-

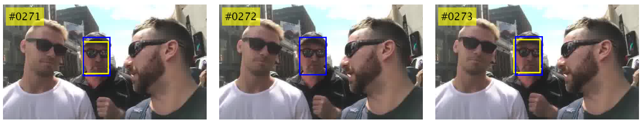

Fig. 5: A typical failure case of the tracker built by concatenating face detection and face recognition (DVNet). The tracker ignores spatio-temporal information in videos and fails when the detector misses the target face, while ECO correctly tracks the target face when it does not move much(best viewed in colour).

ers. A typical example is shown in Fig.5, in which the target face did not move much but DVNet failed to find the face in the middle frame. Last but not least, it is noticeable that comparing to many visual trackers, DVNet is quite slow, running at only approximately 8 FPS in our experiment environment, which has much more computational power that what is typically available on smartphone. Nonetheless, this result also suggests that a hybrid approach that can take advantage of both face detection / verification and visual tracking may provide a more promising alternative to pure visual trackers that do not utilise any prior knowledge about faces.

In the success plot, an interesting observation is that the top 5 trackers all employ deep features. This suggests that meaningful features extracted by deep learning play a key role in mobile face tracking, which is also observed in a recent survey of different components in object tracking[71]. Among the top 5 trackers, MDNet employs a CNN pretrained on a dataset tailored for tracking, while ECO, CCOT and CREST adopt pre-trained VGGNet[61] as a feature extractor. Another observation is the online adaption of the model is extremely important in mobile face tracking. ECO is a CF-based tracker that learns a factorised convolution operator over time for feature fusion from different layers of VGGNet. Similarly, MDNet, a DL-based tracker, finetunes the domain-specific branches during tracking. On the contrary, SiamFC, which employs fixed CNN weights during tracking, performs unfavourably in our case. This shows that online model adaption during tracking is crucial for the tracker to be able to handle appearance changes that are common in mobile tracking scenarios. This can further be support by the performance of DCFNet, which uses similar CNN structure to SiamFC but applies online training during tracking, and it also achieved better performance than that of SimaFC. Similar trends can also be observed in the precision plot. Therefore, a potential direction of further investigation in mobile face tracking could be to develop effective online training strategies.

Another interesting observation is that DVNet only outperforms other best-performing trackers by a small margin despite it was tweaked to output $[0,0,0,0]$ when the target face disappears while all other trackers are forced to predict a (inevitably erroneous) location in such cases. This suggests when the target face re-enters the view, some state-of-the-art object trackers,*e.g*. ECO and MDNet, are still able to re-locate the target.

5.2 Attribute-wise Performance

Fig. 6 shows the performance of 23 evaluated trackers in terms of 9 attributes defined in Table 4. When there are illumination variations(IV), ECO and MDNet perform better than DVNet. This further demonstrates the disadvantage of trivial approach as it ignores spatio-temporal information and thus is unable to consistently locate the target over time. Similar trends can be observed when in-plane rotation (IPR) occurs, where the target may not move much but the face detection cannot utilise the location information from the previous frame. On the other hand, the use of detection algorithm allows DVNet to perform favourably in SV, OCC and OPR because occlusions and the scale and aspect ratio changes can be better handled by numerous region proposal in the detector. This suggests it can be an potential direction to effectively integrate face detection into visual trackers such that they can make complementary learners in mobile face tracking.

5.3 Speed Comparisons

Fig. 3 demonstrates the frame per second (FPS) of 23 evaluated trackers. All trackers are evaluated on an Ubuntu desktop with an Intel(R) Core(TM)i7-7700 3.60GHz CPU and a GeForce GTX 1060 GPU with 3GB memory for fair comparison. Notice that CF-based trackers using hand-engineered features are clearly more computationally efficient than those using deep features. ECO-HC, which uses HOG features within the ECO tracker, strikes a good trade off in terms of FPS and AUC. Most DL-based trackers are much slower even on GPU. IBCCF and CCOT are the slowest trackers that run at less than 1 FPS. Among DL-based trackers, DCFNet shows a better trade off between speed and accuracy since it employs an efficient online CNN training scheme.

6 Conclusion

Mobile face tracking has a multitude of real-world application but it has been largely neglected in the literature. This work aimed to call for more research efforts into mobile face tracking. To this end, we proposed iBUG MobiFace, the first mobile face tracking benchmark with a variety of tracking challenges in real-world mobile scenarios. We surveyed the current state-of-the-art deep learning-based and correlation filter-based object trackers. We also carried out

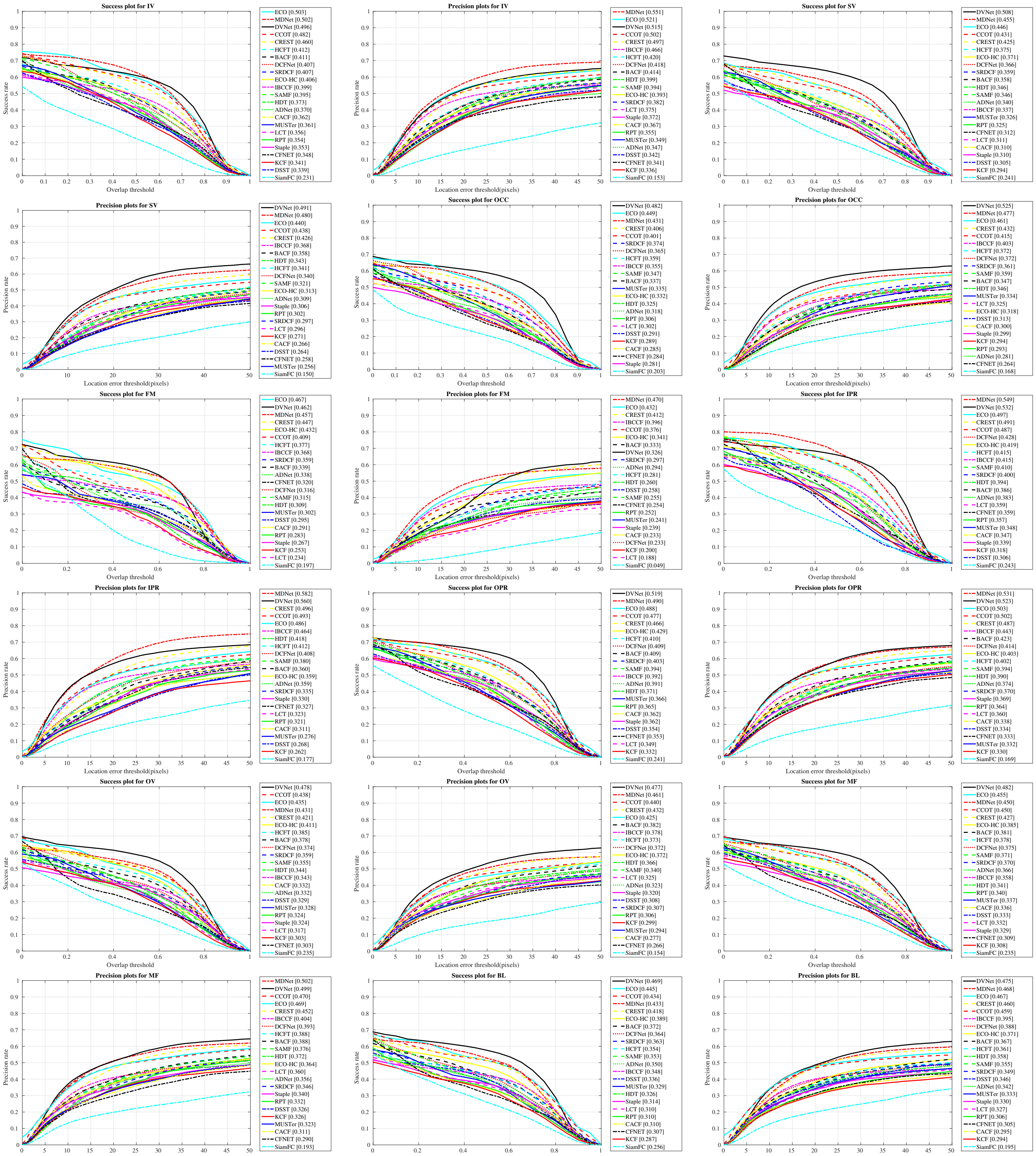

Fig. 6: Attribute-wise performance of 23 evaluated trackers on MobiFace with success plot and precision plot. For success plot, the tracker names are shown with corresponding AUC. For precision plot, the precision at 20 pixel threshold is shown. The 9 attributes are described in Table 4

large scale experiments to evaluate their performance on the proposed benchmark. The evaluation results show that the problem of mobile face tracking is largely unsolved. Both the trivial approach of combining face detection and verification and various visual trackers we covered in the survey failed to achieve a satisfactory performance on our benchmark dataset. In fact, the performance of the state-of-the-art object trackers dropped significantly in this scenario. Nonetheless, a hybrid method that unifies face detection, verification, and visual tracking into a single framework may be a promising direction. A closer inspection also suggests that Deep CNN features could play a key part in top-performing trackers and online adaptation is necessary for trackers to understand the context. Efficient online learning strategies may help the deep learning-based trackers to strike a good balance between speed and accuracy.

References

1. Akin, O., Erdem, E., Erdem, A., Mikolajczyk, K.: Deformable part-based tracking by coupled global and local correlation filters. Journal of Visual Communication and Image Representation **38**, 763 – 774 (2016). DOI https://doi.org/10.1016/j.jvcir.2016.04.018
2. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.: Staple: Complementary Learners for Real-Time Tracking. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1401–1409 (2015). DOI 10.1109/CVPR.2016.156. URL `http://ieeexplore.ieee.org/document/7780525/`
3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. arXiv preprint arXiv:1606.09549 (2016)
4. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2544–2550 (2010). DOI 10.1109/CVPR.2010.5539960
5. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a ”siamese” time delay neural network. In: J.D. Cowan, G. Tesauro, J. Alspector (eds.) Advances in Neural Information Processing Systems 6, pp. 737–744. Morgan-Kaufmann (1994)
6. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional neural network committees for handwritten character classification. In: 2011 International Conference on Document Analysis and Recognition, pp. 1135–1139 (2011). DOI 10.1109/ICDAR.2011.229
7. Cui, Z., Xiao, S., Feng, J., Yan, S.: Recurrently target-attending tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1449–1458 (2016). DOI 10.1109/CVPR.2016.161
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 1, pp. 886–893 vol. 1 (2005). DOI 10.1109/CVPR.2005.177
9. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: Efficient Convolution Operators for Tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6931–6939. IEEE (2017). DOI 10.1109/CVPR.2017.733. URL `http://ieeexplore.ieee.org/document/8100216/`
10. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press (2014)
11. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Learning Spatially Regularized Correlation Filters for Visual Tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4310–4318. IEEE (2015). DOI 10.1109/ICCV.2015.490. URL `http://ieeexplore.ieee.org/document/7410847/`
12. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Discriminative Scale Space Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(8), 1561–1575 (2017). DOI 10.1109/TPAMI.2016.2609928
13. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 621–629 (2015). DOI 10.1109/ICCVW.2015.84
14. Danelljan, M., Khan, F.S., Felsberg, M., van de Weijer, J.: Adaptive Color Attributes for Real-Time Visual Tracking. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1090–1097. IEEE (2014). DOI 10.1109/CVPR.2014.143
15. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: B. Leibe, J. Matas, N. Sebe, M. Welling (eds.) European Conference on Computer Vision, vol. 9909 LNCS, pp. 472–488. Springer International Publishing, Cham (2016). DOI 10.1007/978-3-319-46454-1_29. URL `https://doi.org/10.1007/978-3-319-46454-1{_}29`
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
17. Fan, H., Ling, H.: Sanet: Structure-aware network for visual tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2217–2224 (2017). DOI 10.1109/CVPRW.2017.275
18. Galoogahi, H.K., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1144–1152 (2017). DOI 10.1109/ICCV.2017.129
19. Galoogahi, H.K., Sim, T., Lucey, S.: Correlation filters with limited boundaries. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4630–4638 (2015). DOI 10.1109/CVPR.2015.7299094
20. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587 (2014)
21. Gundogdu, E., Alatan, A.A.: Good features to correlate for visual tracking (2017)
22. Hain, R., Kähler, C.J., Tropea, C.: Comparison of ccd, cmos and intensified cameras. Experiments in fluids **42**(3), 403–411 (2007)
23. Hansen, T.R., Eriksson, E., Lykke-Olesen, A.: Use your head: Exploring face tracking for mobile interaction. In: CHI ’06 Extended Abstracts on Human Factors in Computing Systems, CHI EA ’06, pp. 845–850. ACM, New York, NY, USA (2006). DOI 10.1145/1125451.1125617
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
25. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV, pp. 702–715. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-33765-9_50. URL `https://doi.org/10.1007/978-3-642-33765-9{_}50`
26. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3), 583–596 (2015)

27. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
28. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: Proceedings of the 32nd International Conference on Machine Learning, 2015, Lille, France, 6-11 July 2015 (2015)
29. Hong, Z., Zhe Chen, Wang, C., Mei, X., Prokhorov, D., Tao, D.: MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 749–758. IEEE (2015). DOI 10.1109/CVPR.2015.7298675. URL `http://ieeexplore.ieee.org/document/7298675/`
30. Huang, C., Lucey, S., Ramanan, D.: Learning policies for adaptive tracking with deep feature cascades. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 105–114 (2017). DOI 10.1109/ICCV.2017.21
31. Huang, D., Luo, L., Chen, Z., Wen, M., Zhang, C.: Applying detection proposals to visual tracking for scale and aspect ratio adaptability. International Journal of Computer Vision **122**(3), 524–541 (2017). DOI 10.1007/s11263-016-0974-6. URL `https://doi.org/10.1007/s11263-016-0974-6`
32. Jain, V., Learned-Miller, E.: Fddb: A benchmark for face detection in unconstrained settings. Tech. Rep. UM-CS-2010-009, University of Massachusetts, Amherst (2010)
33. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., Vojir, T., Hager, G., Lukezic, A., Eldesokey, A., Fernandez, G.: The visual object tracking vot2017 challenge results. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
34. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Čehovin, L., Fernandez, G., Vojir, T., Häger, G., Nebehay, G., Pflugfelder, R., Gupta, A., Bibi, A., Lukežič, A., Garcia-Martin, A., Saffari, A., Petrosino, A., Montero, A.S., Varfolomieiev, A., Baskurt, A., Zhao, B., Ghanem, B., Martinez, B., Lee, B., Han, B., Wang, C., Garcia, C., Zhang, C., Schmid, C., Tao, D., Kim, D., Huang, D., Prokhorov, D., Du, D., Yeung, D.Y., Ribeiro, E., Khan, F.S., Porikli, F., Bunyak, F., Zhu, G., Seetharaman, G., Kieritz, H., Yau, H.T., Li, H., Qi, H., Bischof, H., Possegger, H., Lee, H., Nam, H., Bogun, I., Jeong, J.c., Cho, J.i., Lee, J.Y., Zhu, J., Shi, J., Li, J., Jia, J., Feng, J., Gao, J., Choi, J.Y., Kim, J.W., Lang, J., Martinez, J.M., Choi, J., Xing, J., Xue, K., Palaniappan, K., Lebeda, K., Alahari, K., Gao, K., Yun, K., Wong, K.H., Luo, L., Ma, L., Ke, L., Wen, L., Bertinetto, L., Pootschi, M., Maresca, M., Danelljan, M., Wen, M., Zhang, M., Arens, M., Valstar, M., Tang, M., Chang, M.C., Khan, M.H., Fan, N., Wang, N., Miksik, O., Torr, P., Wang, Q., Martin-Nieto, R., Pelapur, R., Bowden, R., Laganiere, R., Moujtahid, S., Hare, S., Hadfield, S., Lyu, S., Li, S., Zhu, S.C., Becker, S., Duffner, S., Hicks, S.L., Golodetz, S., Choi, S., Wu, T., Mauthner, T., Pridmore, T., Hu, W., Hübner, W., Wang, X., Li, X., Shi, X., Zhao, X., Mei, X., Shizeng, Y., Hua, Y., Li, Y., Lu, Y., Li, Y., Chen, Z., Huang, Z., Chen, Z., Zhang, Z., He, Z.: The Visual Object Tracking VOT2015 challenge results. In: Visual Object Tracking Workshop 2015 at ICCV2015 (2015)
35. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(11), 2137–2155 (2016). DOI 10.1109/TPAMI.2016.2516982
36. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Čehovin, L., Nebehay, G., Vojíř, T., Fernández, G., Lukežič, A., Dimitriev, A., Petrosino, A., Saffari, A., Li, B., Han, B., Heng, C., Garcia, C., Pangeršič, D., Häger, G., Khan, F.S., Oven, F., Possegger, H., Bischof, H., Nam, H., Zhu, J., Li, J., Choi, J.Y., Choi, J.W., Henriques, J.F., van de Weijer, J., Batista, J., Lebeda, K., Öfjäll, K., Yi, K.M., Qin, L., Wen, L., Maresca, M.E., Danelljan, M., Felsberg, M., Cheng, M.M., Torr, P., Huang, Q., Bowden, R., Hare, S., Lim, S.Y., Hong, S., Liao, S., Hadfield, S., Li, S.Z., Duffner, S., Golodetz, S., Mauthner, T., Vineet, V., Lin, W., Li, Y., Qi, Y., Lei, Z., Niu, Z.H.: The visual object tracking vot2014 challenge results. In: L. Agapito, M.M. Bronstein, C. Rother (eds.) Computer Vision - ECCV 2014 Workshops, pp. 191–217. Springer International Publishing, Cham (2015)
37. Kumar, B.V., Mahalanobis, A., Juday, R.D.: Correlation pattern recognition. Cambridge University Press (2005)
38. Li, F., Yao, Y., Li, P., Zhang, D., Zuo, W., Yang, M.H.: Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
39. Li, H., Wu, H., Zhang, H., Lin, S., Luo, X., Wang, R.: Distortion-aware correlation tracking. IEEE Transactions on Image Processing **26**(11), 5421–5434 (2017). DOI 10.1109/TIP.2017.2740119
40. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: L. Agapito, M.M. Bronstein, C. Rother (eds.) Computer Vision - ECCV 2014 Workshops, pp. 254–265. Springer International Publishing, Cham (2015)
41. Li, Y., Zhu, J., Hoi, S.C.H.: Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 353–361 (2015). DOI 10.1109/CVPR.2015.7298632
42. Liu, T., Wang, G., Yang, Q.: Real-time part-based visual tracking via adaptive correlation filters. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4902–4912 (2015). DOI 10.1109/CVPR.2015.7299124
43. Liwicki, S., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. IEEE Transactions on Neural Networks and Learning Systems **23**, 1624–1636 (2012)
44. Liwicki, S., Zafeiriou, S., Tzimiropoulos, G., Pantic, M.: Fast and robust appearance-based tracking. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (FG'11), pp. 507–513. Santa Barbara, CA, USA (2011)
45. Lukežič, A., Zajc, L.Č., Kristan, M.: Deformable parts correlation filters for robust visual tracking. IEEE transactions on cybernetics (2017)
46. Lukežic, A., Vojír, T., Zajc, L.C., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4847–4856 (2017). DOI 10.1109/CVPR.2017.515
47. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical Convolutional Features for Visual Tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3074–3082. IEEE (2015). DOI 10.1109/ICCV.2015.352. URL `http://ieeexplore.ieee.org/document/7410709/`
48. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Adaptive Correlation Filters with Long-Term and Short-Term Memory for Object Tracking (2017). URL `http://arxiv.org/abs/1707.02309`
49. Ma, C., Xu, Y., Ni, B., Yang, X.: When correlation filters meet convolutional neural networks for visual tracking. IEEE Signal Processing Letters **23**(10), 1454–1458 (2016). DOI 10.1109/LSP.2016.2601691
50. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5388–5396 (2015). DOI 10.1109/CVPR.2015.7299177
51. Mahbub, U., Sarkar, S., Patel, V.M., Chellappa, R.: Active user authentication for smartphones: A challenge data set and benchmark results. In: 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp. 1–8 (2016). DOI 10.1109/BTAS.2016.7791155
52. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)

53. Mueller, M., Smith, N., Ghanem, B.: Context-aware correlation filter tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1387–1395 (2017). DOI 10.1109/CVPR.2017.152
54. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. CoRR **abs/1510.07945** (2015)
55. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.H.: Hedged Deep Tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016). DOI 10.1109/CVPR.2016.466
56. Qiang Wang Jin Gao, J.X.M.Z.W.H.: Dcfnet: Discriminant correlation filters network for visual tracking. arXiv preprint arXiv:1704.04057 (2017)
57. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788 (2016)
58. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91–99 (2015)
59. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815–823 (2015)
60. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Driessche, G.v.d., Graepel, T., Hassabis, D.: Mastering the game of go without human knowledge. Nature **550**(7676), 354–359 (2017). DOI 10.1038/nature24270. URL `http:https://doi.org/10.1038/nature24270`
61. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
62. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(7), 1442–1468 (2014). DOI 10.1109/TPAMI.2013.230
63. Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R., Yang, M.H.: Crest: Convolutional residual learning for visual tracking. In: IEEE International Conference on Computer Vision, pp. 2555 – 2564 (2017)
64. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
65. Tang, M., Feng, J.: Multi-kernel correlation filter for visual tracking. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3038–3046 (2015). DOI 10.1109/ICCV.2015.348
66. Tao, R., Gavves, E., Smeulders, A.W.M.: Siamese instance search for tracking. In: CVPR (2016)
67. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
68. Vedaldi, A., Lenc, K.: Matconvnet – convolutional neural networks for matlab. In: Proceeding of the ACM Int. Conf. on Multimedia (2015)
69. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: IEEE International Conference on Computer Vision (ICCV) (2015)
70. Wang, L., Ouyang, W., Wang, X., Lu, H.: Stct: Sequentially training convolutional networks for visual tracking. In: CVPR (2016)
71. Wang, N., Shi, J., Yeung, D.Y., Jia, J.: Understanding and diagnosing visual tracking systems. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 3101–3109 (2015). DOI 10.1109/ICCV.2015.355
72. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 809–817. Curran Associates, Inc. (2013)
73. Wang, S., Xiong, X., Xu, Y., Wang, C., Zhang, W., Dai, X., Zhang, D.: Face-tracking as an augmented input in video games: enhancing presence, role-playing and control. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 1097–1106. ACM (2006)
74. van de Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Transactions on Image Processing **18**(7), 1512–1523 (2009). DOI 10.1109/TIP.2009.2019809
75. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
76. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015). DOI 10.1109/TPAMI.2014.2388226
77. Yun, S., Choi, J., Yoo, Y., Yun, K., Young Choi, J.: Action-decision networks for visual tracking with deep reinforcement learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
78. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.H.: Fast Visual Tracking via Dense Spatio-temporal Context Learning. In: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds.) Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V, pp. 127–141. Springer International Publishing, Cham (2014). DOI 10.1007/978-3-319-10602-1_9. URL `https://doi.org/10.1007/978-3-319-10602-1{_}9`
79. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (2016). DOI 10.1109/LSP.2016.2603342
80. Zhang, T., Xu, C., Yang, M.H.: Multi-task Correlation Particle Filter for Robust Object Tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4819–4827. IEEE (2017). DOI 10.1109/CVPR.2017.512. URL `http://ieeexplore.ieee.org/document/8099995/`
81. Zhu, G., Wang, J., Wu, Y., Lu, H.: Collaborative correlation tracking. In: Proceedings of the British Machine Vision Conference (BMVC), pp. 184.1–184.12. BMVA Press (2015). DOI 10.5244/C.29.184. URL `https://dx.doi.org/10.5244/C.29.184`
82. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds.) Computer Vision – ECCV 2014, pp. 391–405. Springer International Publishing, Cham (2014)
83. Čehovin, L., Leonardis, A., Kristan, M.: Visual object tracking performance measures revisited. IEEE Transactions on Image Processing **25**(3), 1261–1274 (2016). DOI 10.1109/TIP.2016.2520370