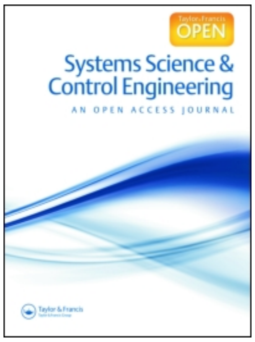

Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) 2164-2583 (Online) Journal homepage: https://www.tandfonline.com/loi/tssc20

Visual object tracking: in the simultaneous presence of scale variation and occlusion

Fan Feng, Bo Shen & Hongjian Liu

To cite this article: Fan Feng, Bo Shen & Hongjian Liu (2018) Visual object tracking: in the simultaneous presence of scale variation and occlusion, Systems Science & Control Engineering, 6:1, 456-466, DOI: 10.1080/21642583.2018.1536899

To link to this article: https://doi.org/10.1080/21642583.2018.1536899

Published online: 25 Oct 2018.

SYSTEMS SCIENCE & CONTROL ENGINEERING: AN OPEN ACCESS JOURNAL
2018, VOL. 6, NO. 1, 456–466
https://doi.org/10.1080/21642583.2018.1536899

Taylor & Francis
Taylor & Francis Group

Visual object tracking: in the simultaneous presence of scale variation and occlusion

Fan Feng[a,b], Bo Shen[a,b] and Hongjian Liu[c]

[a]College of Information Science and Technology, Donghua University, Shanghai, People's Republic of China; [b]Engineering Research Center of Digitalized Textile and Fashion Technology, Ministry of Education, Shanghai, People's Republic of China; [c]School of Mathematics and Physics, Anhui Polytechnic University, Wuhu, Shanghai, People's Republic of China

ABSTRACT
Visual object tracking is a challenging task when the object appearance changes caused by the scale variation and the occlusion. In this paper, an object tracking algorithm is proposed which is capable of dealing with the case that the scale variation and the occlusion occur simultaneously. A kernelized correlation filter (KCF) is first learned to obtain the correlation response, whose maximum value denotes the optimal object location. In order to represent the sample better, the convolutional features are extracted from a pre-trained convolutional neural networks (CNNs). Then, the strategy of scale adaption is used to estimate the object scale during the tracking process. Subsequently, a novel re-detection model is proposed by using a support vector machine (SVM) classifier to re-find the object when the occlusion occurs. The comparison experiments are implemented on the object tracking benchmark (OTB) and the results demonstrate that the proposed tracking algorithm outperforms other state-of-the-art ones in terms of precision and success rate.

ARTICLE HISTORY
Received 28 June 2018
Accepted 13 October 2018

KEYWORDS
Neural networks; support vector machines; image processing and vision; tracking

1. Introduction

Visual object tracking is a challenging task in computer vision and it has wide applications such as video surveillance, human computer interaction, medical imaging and robotics. The basic idea of visual object tracking is to track an object in the sequential frames, whose central position and bounding box are given in the first frame. It should be mentioned that, for the case when the object appearance changes, there has not yet been an effective tracking method which is suitable for all the scenarios. In this work, we focus mainly on the scenarios of the scale variation and the occlusion, and desire to achieve the tracking task in these scenarios.

So far, for the object tracking problem without considering the scale variation and the occlusion, there have been a number of classical tracking algorithms available in the existing literature. Among various tracking algorithms, the discriminative model-based approach is representative that has received considerable attention, see e.g. Hare et al. (2016), Lucey (2008), Wang, Hua, and Han (2010), Kalal, Matas, and Mikolajczyk (2010), Zhang, Zhang, and Yang (2012), and Wang, Chen, Xu, and Yang (2015). As one of the discriminative model-based approaches, the correlation filter-based tracking method has been proposed in Bolme, Beveridge, Draper, and Lui (2010). The correlation filter is capable of producing a correlation peak at the object although it gets a low response in the background region. Due to its fast tracking speed, the correlation filter-based tracking method has attracted increasing attention from researchers in this area. Further, much efforts have been made on the improvement of the correlation filter-based tracking method. For example, the kernelized correlation filter (KCF) (Henriques, Caseiro, Martins, & Batista, 2015) has been trained by minimizing the squared error over the samples with a cyclic structure and their two-dimensional Gaussian labels. The circulant structure is exploited to obtain more positive and negative samples for the classifier training without sacrificing tracking speed. Kernel function is employed to map each sample to a nonlinear space. In the new frame, each sample can produce a response value through the learned KCF, and the maximum value indicates the optimal object location. In Danelljan, Shahbaz Khan, and Van de Weijer (2014), the colour-attributes have been employed to represent object appearance in order to improve the tracking robustness. In Danelljan, Hager, Shahbaz, and Felsberg (2015), unwanted boundary effects introduced by the periodic assumption have been relieved to improve the quality of the tracking model.

CONTACT B. Shen bo.shen@dhu.edu.cn, shenbodh@gmail.com

Except for the correlation filter-based tracking methods, the convolutional neural networks (CNNs) based tracking algorithms have also been developed rapidly. CNNs, as a powerful tool, have been applied in a number of visual tasks such as object detection (Girshick, 2015; Zeng, Wang, Zhang, Liu, & Alsaadi, 2016), face recognition (Parkhi, Vedaldi, & Zisserman, 2015; Zeng et al., 2017), semantic segmentation (Long, Shelhamer, & Darrell, 2015) and image classification (Krizhevsky, Sutskever, & Hinton, 2012). In the framework of CNNs, images need to be pre-processed before being input to CNNs for evaluation, and the structure of CNNs usually contains convolutional, pooling and fully connected layers. Generally, there are mainly two kinds of tracking methods based on CNNs: the tracking methods based on the dedicated CNNs and the tracking methods by using convolutional features. For the former, the characteristics of multi-domain or siamese structure have been employed and the dedicated CNNs based object tracking algorithm has been developed, see, e.g. Nam and Han (2016), Bertinetto, Valmadre, Henriques, Vedaldi, and Torr (2016) and Held, Thrun, and Savarese (2016). In the latter, the features have been extracted from the convolutional layers of the pre-trained CNNs in order to improve the object tracking performance. Based on this idea, a variety of trackers based on the convolutional features have been proposed. For example, the spatially regularized discriminative correlation filter with deep features (deepSRDCF) based tracker has been proposed in Danelljan, Hager, Shahbaz Khan, and Felsberg (2015), where the features of the first convolutional layer are utilized and the principal component analysis (PCA) is employed to reduce the feature dimensionality. In Qi et al. (2016), the hedged deep tracking (HDT) algorithm has been designed by making full use of the features from different layers and hedging the weak trackers into a strong tracker.

With respect to the case of scale variation, there have also been a large number of effective tracking methods available in the existing literature. For example, the discriminative scale space-based tracker (DSST) has been proposed in Danelljan, Häger, Khan, and Felsberg (2014), where the translation filter and the scale filter are trained to estimate the object position and object scale, respectively. In Li and Zhu (2014), the responses of seven different size samples have been calculated to determine the most suitable object scale. The object scale has been adjusted in Zhang, Zhang, Liu, Zhang, and Yang (2014) by updating the scale parameter in the weight function and the spatio-temporal context tracker (STC) has been proposed. In Montero, Lang, and Laganire (2015), an adjustable Gaussian window function and a keypoint-based model for scale estimation are utilized to deal with the fixed size limitation.

It is well known that the tracking performance is easily affected when the object is sheltered. When occlusion occurs, the appearance of the object changes, which leads to tracking deviations or failure. Therefore, it is crucial for the trackers to re-find the object under the occlusion situation. For this purpose, several object tracking approaches have been proposed. For example, the multi-store tracker (MUSTer) has been proposed in Hong et al. (2015), where the tracking task is divided into short-term and long-term memory. In short-term memory, an integrated correlation filter has been trained while, in long-term memory, more additional information has been provided by using the method based on keypoint matching and random sample consensus (RANSAC) estimation. In Wang, Liu, and Huang (2017), a multimodal object detection technique has been exploited to prevent model drift introduced by similar objects or background noises and the large margin object tracking method has been proposed with the help of circulant feature maps.

In order to take into account both scale variation and occlusion, the long-term correlation tracker (LCT) has been proposed in Ma, Yang, Zhang, and Yang (2015). The proposed LCT is decomposed into three parts: location estimation, scale estimation and location re-detection. For the location estimation, the KCF has been utilized and the scale filter (Danelljan et al., 2014) has been exploited to estimate the object scale. In order to re-detect the object location, the online random fern classifier (Kalal, Mikolajczyk, & Matas, 2012) has been employed. However, it should be pointed out that there have still been much improvement space in the LCT mentioned above. For example, in the location estimation in the LCT, the KCF is trained by using the histogram of orientation gradients (HOG) features which are not robust enough to represent samples. Since another KCF needs to be trained to produce correlation response, the calculation of the object appearance is time-consuming. The online random fern classifier is not strong enough to find the optimal object from the plentiful candidate samples.

In this paper, a visual object tracking algorithm is proposed to resolve the problems induced by the scale variation and occlusion simultaneously. The main contributions of this paper are summarized as follows: *(1) A KCF is learned and the convolutional features are extracted to strengthen the ability to represent the sample; (2) a scale adaption strategy is employed to estimate the object scale; and (3) a novel re-detection model is proposed by combining the the peak to sidelobe ratio (PSR) and support vector machine (SVM) classifier to re-find the object when occlusion* *occurs*. In contrast with other tracking approaches, the proposed one seems to be more favorable in dealing with occlusion owning to its distinctive advantages in simplified occlusion judgment and reliable classifier. In addition, the multilayer convolutional features are utilized which improves the performance of KCF. In order to verify the effectiveness of the proposed algorithm, some comparison experiments are implemented. The experimental results show that the proposed tracking algorithm outperforms other existing algorithms in terms of the precision and the success rate.

Notation Throughout this paper, $\mathcal{F}$ represents the discrete fourier transform (DFT). The symbol '$\odot$' denotes the element-wise multiplication, '*' indicates the complex conjugate, and '$\cdot$' is the inner product. The superscript 'T' refers to matrix transposition.

2. Related works

In this section, we discuss the tracking methods closely related to this work: (i) tracking by correlation filter, (ii) tracking by convolutional features and (iii) tracking by re-detection model. Correlation filter-based tracking approach belongs to the discriminative model-based method, where the classifier is trained to distinguish the object from the background. Representation scheme is one of the major components in any visual tracker, and convolutional features are used to represent sample, which have a reliable performance. As for model integration, the re-detection model-based tracking algorithm is capable of switching model when the tracking result is not ideal.

Tracking by correlation filter. Visual object tracking algorithms based on correlation filter have become a development tendency in the last few years. Among them, the circulant matrix of the sample is used to reduce computational complexity of the dense sampling in KCF-based tracker (Henriques et al., 2015). The KCF is trained by minimizing the squared error over the samples with a cyclic structure and their two-dimensional Gaussian labels. When the next frame comes, the correlation response is obtained by the new samples with a cyclic structure through the learned KCF. Each sample can produce a response value and the maximum value indicates the optimal object location. The DSST-based tracker (Danelljan et al., 2014) can deal with the case of scale variation, in which the translation filter and the scale filter are exploited to estimate location and scale of the object, respectively. In the step of scale estimation, each sample of different sizes can produce a response value, and the best scale is also indicated by the maximum response value.

Tracking by convolutional features. Visual object tracking methods based on convolutional features have shown the state-of-the-art performance. The parameters of convolutional layers, which are extracted from the pre-trained VGGNet, is used to represent sample features in hierarchical correlation filters (HCF) tracker (Ma, Huang, Yang, & Yang, 2015). The features of latter convolutional layer contain more semantic information and are robust to the changing object appearance, which are used to distinguish the object from the backgrounds, while the ones of the earlier convolutional layer provide more location information and are convenient for the object location. In order to strengthen the robustness of the features, the parameters of three convolutional layers: conv5-4, conv4-4 and conv3-4 are utilized to constitute sample features. Three KCFs are trained via the features of each layer and then three-layer correlation responses can be obtained correspondingly. After that, a coarse-to-fine strategy, which follows the order of the correlation responses from conv5-4 layer, conv4-4 layer to conv3-4 layer, is proposed to acquire the accuracy object position.

Tracking by re-detection model. Visual object tracking methods based on re-detection model can evaluate the tracking results and re-find the object if tracking fails such as occlusion. In the LCT (Ma et al., 2015), another KCF is used to obtain the reliable object appearance result and an online random ferns classifier is exploited to redetermine the object location when the object is subject to occlusion.

3. Proposed algorithm

Scale variation and occlusion are two common scenarios during the tracking process. Scale variation means that the ratio of the bounding boxes of the first frame and the current frame is out of the range $[1/\theta, \theta]$, $\theta > 1$ (e.g. $\theta = 2$). Occlusion means that the target is partially or fully occluded. In the OTB, the videos including scale variation and occlusion scenarios account for 55% and 57% of all the test videos, respectively. In the case of scale variation and occlusion, the object is prone to tracking drift even tracking failure.

In order to achieve the reliability of the tracking algorithm, the tracker should be robust to the object appearance changes. For this purpose, we first train a KCF by using the convolutional features to estimate object location. Then, a scale filter is learned by using the HOG features to estimate the object scale. Subsequently, we calculate the PSR of total correlation responses to determine whether the SVM classifier (Cortes & Vapnik, 1995) is used for the object re-detection in case of occlusion. The overall flowchart is illustrated in Figure 1.

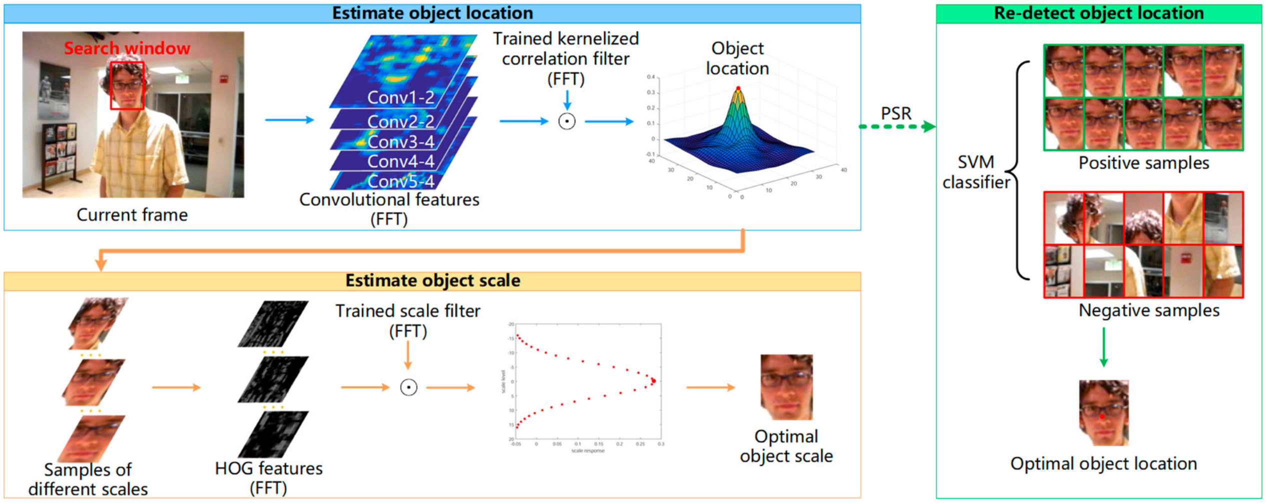

Figure 1. The overall flowchart of the proposed tracking algorithm. The tracking method is divided into three steps: object location estimation, object scale estimation and object location re-detection. PSR is used to determine whether the re-detection model is activated or not.

3.1. Kernelized correlation filter

The aim of the correlation filter is to train a filter h such that the correlation response g and the input image f satisfy $g = f \star h$ where the symbol '$\star$' denotes the correlation operator. According to convolution theorem, the correlation in the time domain can be converted into the element-wise multiplication in the frequency domain as follows:

$$\mathcal{F}(g) = \mathcal{F}(f) \odot \mathcal{F}(h)^{*}. \tag{1}$$

By considering the computational complexity of correlation operation, the element-wise multiplication in the frequency domain is adopted which can improve the tracking speed. The position of the maximum value of the correlation response g is employed to index the object location.

Note that there is a common problem in traditional correlation filter, i.e. training samples are insufficient for the filter training. In order to solve this problem, the dense sampling (Henriques, Caseiro, Martins, & Batista, 2012) has been introduced in the correlation filter. However, the increase of training samples inevitably leads to the increase of the computational complexity. Therefore, in this section, the strategy of cyclic shifts (Henriques et al., 2015) is adopted to simplify sampling and calculation.

In this section, a KCF $\mathbf{w}$ is trained by using an image patch $\mathbf{x}$ of $M \times N$ pixels, which is cut out from the first frame in the video sequences. As mentioned above, we shift the sample $\mathbf{x}$, which is also called the base sample, along the M and N dimensions and generate the cyclic shifts samples. Each sample can be denoted as $\mathbf{x}_{m,n}$, $(m,n) \in \{(1-M)/2, \ldots, -1, 0, 1, \ldots, (M-1)/2\} \times \{(1-N)/2, \ldots, -1, 0, 1, \ldots, (N-1)/2\}$, and we add a Gaussian label $y(m,n) = e^{-(m^2+n^2)/2\sigma^2}$, $y(m,n) \in (0,1]$ to each sample where σ is used to indicate the kernel width. The samples of cyclic shifts and their labels are illustrated in Figure 2. Then, we calculate the KCF $\mathbf{w}$ by minimizing the squared error over samples $\mathbf{x}_{m,n}$ and theirlabels $y(m,n)$ as follows

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{m,n} (\varphi(\mathbf{x}_{m,n}) \cdot \mathbf{w} - y(m,n))^2 + \lambda \|\mathbf{w}\|^2, \tag{2}$$

where λ denotes regularization parameter, $\|\mathbf{w}\|$ represents Euclidean norm of $\mathbf{w}$, and φ is themapping from a linear space to a kernel space which is a nonlinear space.In Williams (2003), the solution $\mathbf{w}$ can be expressed by a linear combination of samples,i.e. $\mathbf{w} = \sum_{m,n} \boldsymbol{\alpha}(m,n)\varphi(\mathbf{x}_{m,n})$, where $\boldsymbol{\alpha}$ is the coefficient. According to Rifkin, Yeo, and Poggio (2003), thecoefficient $\boldsymbol{\alpha}$ can be calculated as

$$\mathcal{F}(\boldsymbol{\alpha}) = \frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\varphi(\mathbf{x}) \cdot \varphi(\mathbf{x})) + \lambda}, \tag{3}$$

where $\mathbf{y} = \{y(m,n) | (m,n) \in \{(1-M)/2, \ldots, -1, 0, 1, \ldots, (M-1)/2\} \times \{(1-N)/2, \ldots, -1, 0, 1, \ldots, (N-1)/2\}$. $\varphi(\mathbf{x}) \cdot \varphi(\mathbf{x}) = \kappa(\mathbf{x}, \mathbf{x})$, where κ is a kernel function such as linear kernel, polynomial kernel and radial basis function kernel. When the new frame of the video coming, we cut out a imagepatch $\mathbf{z}$ at object position of previous frame with the same size as sample $\mathbf{x}$. Therefore, thefollowing correlation response $\mathbf{f}$ can be obtained through the correlation filter $\mathbf{w}$ and thenew sample $\mathbf{z}$

$$\mathcal{F}(\mathbf{f}) = \mathcal{F}(\boldsymbol{\alpha}) \odot \mathcal{F}(\varphi(\mathbf{z}) \cdot \varphi(\mathbf{x})). \tag{4}$$

Note that $\mathbf{x}$ is the sample of last frame, which is called the trained sample, and $\mathbf{z}$ is the sample of current frame,

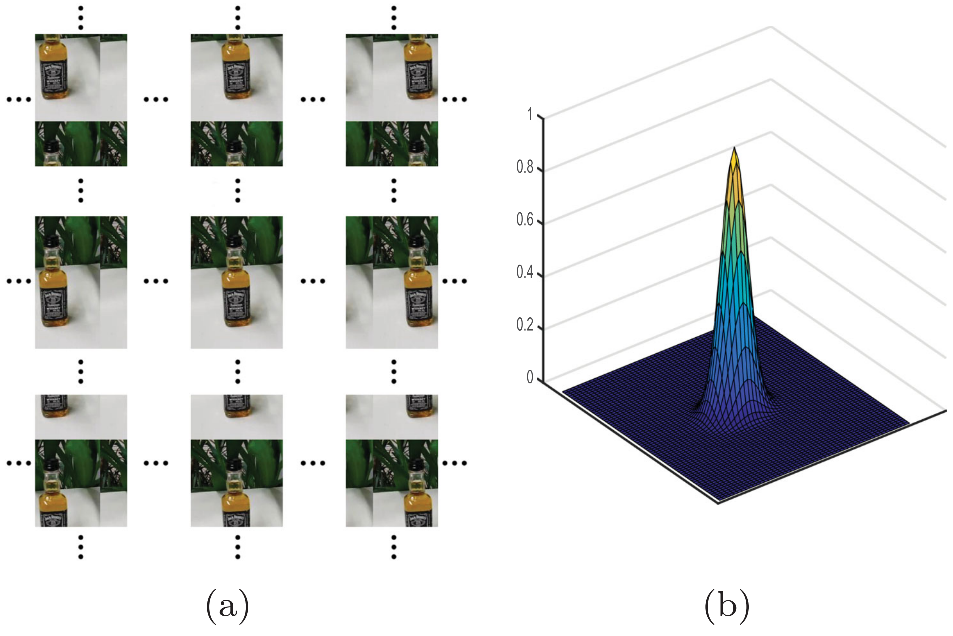

Figure 2. The samples of cyclic shifts and their labels. (a) The base sample is the middle one, and the samples of cyclic shifts are distributed around the base sample. (b) Each sample has a Gaussian label, and the closer the sample of cyclic shifts is to the base sample, the larger the label value is.

which is called the test sample. In general, we employ the trained sample **x** of previous frame to learn the coefficient $\boldsymbol{\alpha}$ and thus obtain the KCF **w**. Then, we use test the sample **z** of current frame to obtain the correlation response **f**. The location of maximum value of the correlation response **f** is the optimal object position in the current frame.

In order to improve the robustness of the tracker, the trained sample **x** and the coefficient $\boldsymbol{\alpha}$ are updated frame by frame according to the following formula

$$\begin{aligned} \mathbf{x}_t &= (1-\beta)\mathbf{x}_{t-1} + \beta\mathbf{x}_t, \\ \boldsymbol{\alpha}_t &= (1-\beta)\boldsymbol{\alpha}_{t-1} + \beta\boldsymbol{\alpha}_t, \end{aligned} \tag{5}$$

where β is a learning rate and t is the index of the current frame. This updating strategy can maintain the memorability of the tracker and thus avoid unexpected interference.

3.2. Convolutional features

In order to achieve more robustness and accurateness of the tracker, the convolutional features is used instead of the raw image or handcrafted features to represent the samples. We employ VGGNet-19 (Simonyan & Zisserman, 2014) which is trained on the ImageNet dataset (Deng et al., 2009). The VGGNet-19 consists of 16 convolutional layers which are divided into 5 groups, 5 pooling layers, 3 fully-connected layers and 1 soft-max layer. The detailed configuration of VGGNet-19 is shown in Table 1.

Table 1. The configuration of VGGNet-19. The convolutional layers are denoted as 'Conv $<$ group number $>$ - $<$ serial number $>$'. Note that maxpool is used in this network, which is a kind of pooling. FC and soft-max indicate fully-connected layer and soft-max layer, respectively.

Group	Layer	Size
	Input RGB image	224 × 224 × 3
Group1	Conv1-1	224 × 224 × 64
	Conv1-2	224 × 224 × 64
	maxpool	112 × 112 × 64
Group2	Conv2-1	112 × 112 × 128
	Conv2-2	112 × 112 × 128
	maxpool	56 × 56 × 128
Group3	Conv3-1	56 × 56 × 256
	Conv3-2	56 × 56 × 256
	Conv3-3	56 × 56 × 256
	Conv3-4	56 × 56 × 256
	maxpool	28 × 28 × 256
Group4	Conv4-1	28 × 28 × 512
	Conv4-2	28 × 28 × 512
	Conv4-3	28 × 28 × 512
	Conv4-4	28 × 28 × 512
	maxpool	14 × 14 × 512
Group5	Conv5-1	14 × 14 × 512
	Conv5-2	14 × 14 × 512
	Conv5-3	14 × 14 × 512
	Conv5-4	14 × 14 × 512
	maxpool	7 × 7 × 512
	FC	1 × 1 × 4096
	FC	1 × 1 × 4096
	FC	1 × 1 × 1000
	soft-max	1 × 1 × 1000

The size of input image in this network is fixed to 224 × 224 pixels and we get rid of the fully-connected layers and soft-max layer to save the time of forward propagation.

Conveniently, the parameters of convolutional layer can represent the image features. The features extracted from the earlier convolutional layer store more location information, which can be utilized to acquire accurate object location, owing to their high spatial resolution. However, the features extracted from the earlier convolutional layer are not robust to appearance changes of the object, because the earlier convolutional layer is far from the classification layer. In contrast, the features extracted from the latter convolutional layer contain more sematic information, which are particularly robust even though the object appearance undergoes serious changes, such as occlusion. However, it should be mentioned that the resolution of the features extracted from the latter convolutional layer is too low to obtain a precise object location. Taking into account the robustness and accuracy, in this section, we employ all the features extracted from last convolutional layers of five groups to represent the image with the hope to achieve a more extraordinary performance.

Different from (Ma et al., 2015), in our work, the image is represented by 5-layer convolutional features extracted from Conv1-2, Conv2-2, Conv3-4, Conv4-4 and Conv5-4, which are illustrated in Figure 1. In the current frame, we first extract 5-layer convolutional features of the trained sample $\mathbf{x}$, and the features for each layer can be denoted as $\mathbf{x}_l$ $(l = 1, 2, \ldots, 5)$. Then every layer KCF $\mathbf{w}_l$ is trained by using the convolutional features $\mathbf{x}_l$ on the corresponding layer. When the new frame comes, we extract 5-layer convolutional features from the test sample $\mathbf{z}$, and each layer features of $\mathbf{z}$ can be denoted as $\mathbf{z}_l$ $(l = 1, 2, \ldots, 5)$. The correlation response $\mathbf{f}_l$ on each layer can be obtained by the obtained $\mathbf{z}_l$ and $\mathbf{w}_l$.

In order to obtain an accurate object location, we add these 5-layer correlation responses together, which can be written as

$$\mathbf{ft} = \sum_{l=1}^{5} \mu_l \mathbf{f}_l, \tag{6}$$

where $\mathbf{ft}$ denotes the total correlation response and μ_l is the weighted value of the corresponding layer. The location of the maximum value of the total correlation response $\mathbf{ft}$ is the optimal object location, which can be expressed as $(\hat{m}, \hat{n}) = \arg\max_{m,n} \mathbf{ft}(m, n)$ where $(\hat{m}, \hat{n})$ indicates the best object location.

3.3. Scale adaption

After obtaining the optimal object location by using the kernelized correlation filter with convolutional features, we are now ready to estimate the object scale. Similar to Danelljan et al. (2014), we adopt the adaptive scale strategy to estimate the object scale. Moreover, in order to simplify the calculation, the HOG features are extracted to represent the sample image instead of the convolutional features.

In the current frame, we first train a scale filter by minimizing the loss over the samples and their labels. When the new frame comes, we collect S (S is an odd number) samples of different sizes at the object location of previous frame, which is illustrated in Figure 1. The size of the raw sample is represented as $P \times Q$, and the sizes of S different samples are represented as $R_r P \times R_r Q$ $(r = 1, 2, \ldots, S)$ where R_r indicates the r-th element in the set R, $R = \{u^v | v = \lfloor -(S-1)/2 \rfloor, \lfloor -(S-3)/2 \rfloor, \ldots, \lfloor \pm(S-S)/2 \rfloor, \ldots, \lfloor (S-3)/2 \rfloor, \lfloor (S-1)/2 \rfloor\}$ and u denotes the scale multiplier. Then we resize all the samples to $P \times R$ again. Finally, S scale responses can be obtained by using S samples of different sizes through the trained scale filter and the optimal object scale can be obtained as follows

$$\hat{r} = \arg\max_{r} (\mathbf{fs}_1, \mathbf{fs}_2, \ldots, \mathbf{fs}_r, \ldots, \mathbf{fs}_S) \tag{7}$$

where $\mathbf{fs}_r$ is the r-th scale response. As such, the optimal object scale can be expressed as $R_{\hat{r}} P \times R_{\hat{r}} Q$. In order to maintain the instantaneity, the scale filter and the trained sample are updated frame by frame. The detailed process can be referred to Danelljan et al. (2014).

3.4. Re-detection model

In order to handle the problem of occlusion, we propose a re-detection model to re-find the object position. The re-detection model is carried out when the tracking result is not ideal. We use PSR of the total correlation response $\mathbf{ft}$ to describe the tracking effect. Before calculating the PSR, we divide the total correlation response $\mathbf{ft}$ into the peak (which is the maximum value of $\mathbf{ft}$) and the sidelobe (which is the rest of $\mathbf{ft}$ except for 11 × 11-pixel area centered around the peak) and the PSR can be calculated as

$$PSR = \frac{m_p - \mu_{sl}}{\sigma_{sl}} \tag{8}$$

where m_p is the peak value, μ_{sl} and σ_{sl} represent the mean and standard deviation of the sidelobe, respectively. Moreover, two thresholds, i.e. τ_a and τ_b are introduced to represent the different tracking states. When $PSR \geq \tau_a$, it means that the tracking result is convincing. When $PSR \leq \tau_b$, it is indicated that the object tracking fails and the occlusion occurs and, in this case, the re-detection model is activated to find the object again.

We assume that the location where the object reappears is near the position of object occlusion. Then, the following SVM classifier (Cortes & Vapnik, 1995) is trained to find the possible object position again

$$\mathbf{g} = \mathbf{w}_s^\top \mathbf{x}_s + b_s \tag{9}$$

where $\mathbf{x}_s$ denotes the candidate samples that are collected in a large search window, $\mathbf{w}_s$ and b_s indicate the weight and bias of the SVM classifier. Here, the sample labels are determined by the Euclidean distance from the samples location to the object location of the previous frame. As such, the optimal object location is indexed by the maximum value of $\mathbf{g}$. Moreover, if $PSR \geq \tau_a$, $\mathbf{w}_s$ and b_s of SVM classifier will be updated.

Remark 1: Note that, in the location re-detection of LCT in (Ma et al., 2015), another KCF is required to be trained to obtain the correlation response and an online random ferns classifier is employed to re-detect the object when the occlusion occurs. It should be pointed out that, due to the training of another KCF, the calculation of the object appearance may be more complicated and time-consuming. In our work, we use the PSR of the total correlation response **ft** to describe the tracking result to save the computation time introduced by training another KCF. Moreover, we use the SVM classifier instead of random ferns classifier to re-find the optimal object sample from the candidate samples.

4. Implementation details

In this section, we present the implementation details of the proposed tracking method and the main steps of the algorithm are illustrated in Algorithm ??. In the design of KCF, the size of the searching window is set to be 1.8 times of object size, and the spatial size of the sample features is fixed to be one sixteenth of the searching window. The bandwidth to generate Gaussian labels is chosen as $\sigma = 0.1$, the regularization parameter in (2) is set as $\lambda = 10^{-4}$ and the learning rate in (5) is set to $\beta = 0.1$. We also add a cosine widow to the sample features to restrain the boundary discontinuities of cyclic operation. For the convolutional features, the parameters μ_l ($l = 1,2,3,4,5$) in (6) are set as 0.0005, 0.001, 0.02, 0.5 and 1.1 for Conv1-2, Conv2-2, Conv3-4, Conv4-4 and Conv5-4, respectively. Because of the pooling operation, the features extracted from earlier convolutional layer have a high resolution, while the features extracted from latter convolutional layer have a low resolution. Therefore we employ bilinear interpolation to adjust the sizes of features from five convolutional layers to be the same. In the step of the scale adaption, we set $S = 33$ and $u = 1.02$. In the re-detection model, the parameters τ_a and τ_b are set as 10 and 4, respectively.

Algorithm 1: Proposed tracking algorithm.

```
Input: Initial object location and scale,
Output: Estimated object state $(x_t^\star, y_t^\star, r_t^\star)$ in frame
        $t$, where $(x_t^\star, y_t^\star)$ denotes the optimal
        object location and $r_t^\star$ denotes the
        optimal object scale.

while t is not the last frame do
    //Estimate object location
    Crop out the sample in frame $t$ centered
      around $(x_{t-1}^\star, y_{t-1}^\star)$ and extract convolutional
      features of different layers;
    Calculate the correlation response $\mathbf{f}_l$ of each
      layer respectively using (4), obtain the total
      correlation response ft using (6);
    Estimate the object location $(\hat{x}_t, \hat{y}_t)$;
    //Estimate object scale
    Collect $S$ samples of different sizes centered
      around $(\hat{x}_t, \hat{y}_t)$, and calculate the scale
      response $\mathbf{fs}_r$;
    Estimate the optimal object scale $\hat{r}_t$ using (7);
    $r_t^\star = \hat{r}_t$;
    //Re-detect object location
    if $PSR \leq \tau_b$ then
        Activate the re-detection model to re-find
          the optimal object location $(x_t^\star, y_t^\star)$;
    else
        $(x_t^\star, y_t^\star) = (\hat{x}_t, \hat{y}_t)$;
    end
    Acquire the estimated object state $(x_t^\star, y_t^\star, r_t^\star)$;
    //Update
    Update $\mathbf{x}$, $\boldsymbol{\alpha}$ using (5);
    Update scale filter and the trained sample;
    if $PSR \geq \tau_a$ then
        Update $\mathbf{w}_s$ and $b_s$ of the SVM classifier;
    end
end
```

5. Experimental results

In this section, the tracking algorithm proposed in this paper is evaluated and the performance of the proposed tracking algorithm is compared with the ones of other algorithms in the existing literature.

The tracking algorithms are evaluated on the object tracking benchmark (OTB) in Wu, Lim, and Yang (2013) including 50 videos called Benchmark-50. The MATLAB platform and the computer configuration on which the tracking algorithms is tested are given as follows: Intel i7-7700 3.60 GHz CPU with 32 GB RAM and MatConvNet Toolbox (Vedaldi & Lenc, 2015) with a NVIDIA GeForce

GTX 1080 GPU (which is used to accelerate the computation of forward propagation in CNNs). Our tracking algorithm is compared with other 10 state-of-the-art methods which can be divided into three categories: (i) the deep learning tracker DLT (Wang & Yeung, 2013), (ii) the correlation filter trackers including CSK (Henriques et al., 2012), KCF (Henriques et al., 2015), DSST (Danelljan et al., 2014) and STC (Zhang et al., 2014), and (iii) the trackers with single or multiple online classifiers CT (Zhang et al., 2012), Struck (Hare et al., 2016), MIL (Babenko, Yang, & Belongie, 2011), SCM (Zhong, Lu, & Yang, 2014) and TLD (Kalal et al., 2012).

During the evaluation process, the main indices we consider are the precision and success rate. For the

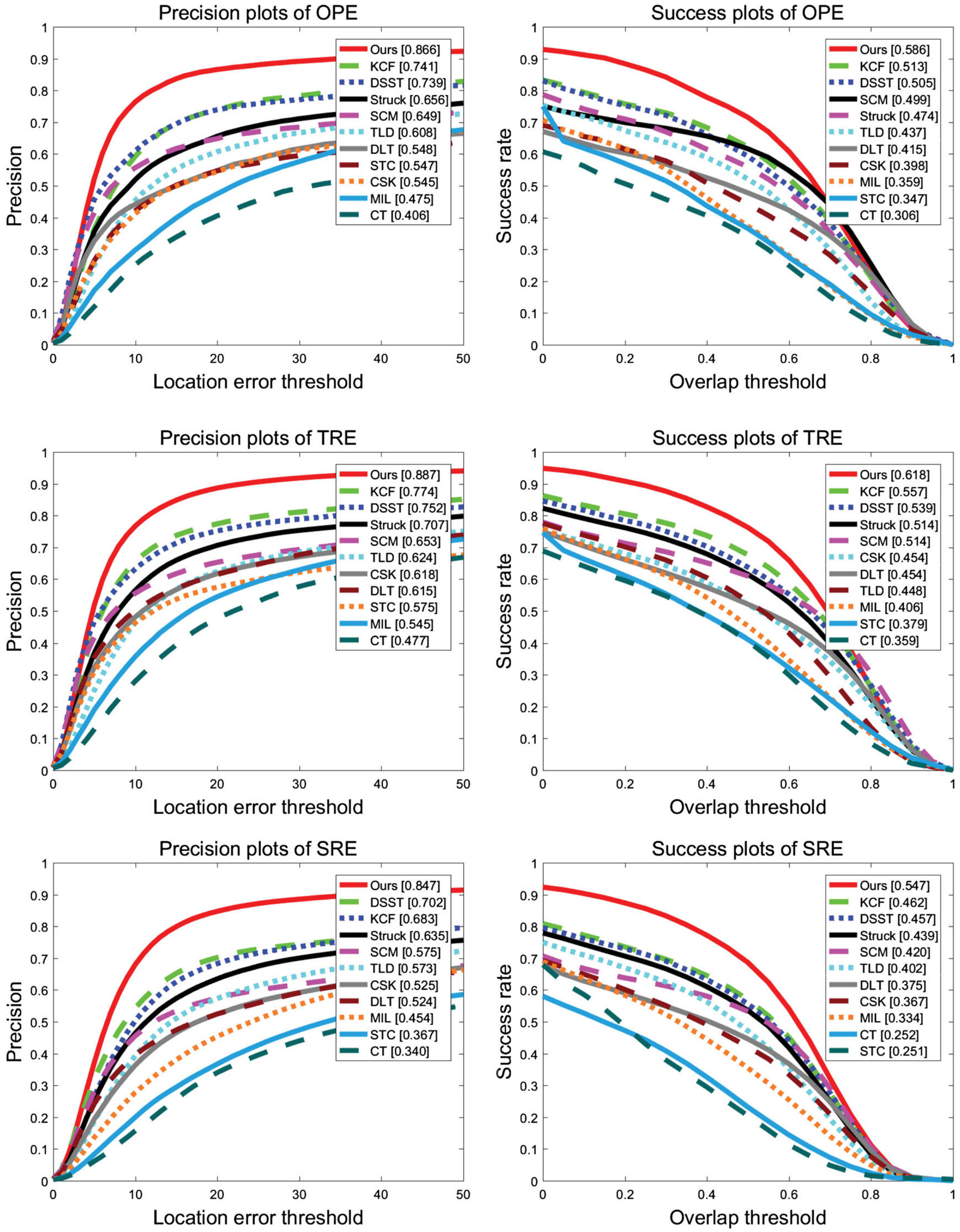

Figure 3. Precision and success plots over 50 test video sequences using OPE, TRE and SRE. The threshold t_p in precision plot is set to 20 pixels and the ranking results in success plot is uesd by the AUC.

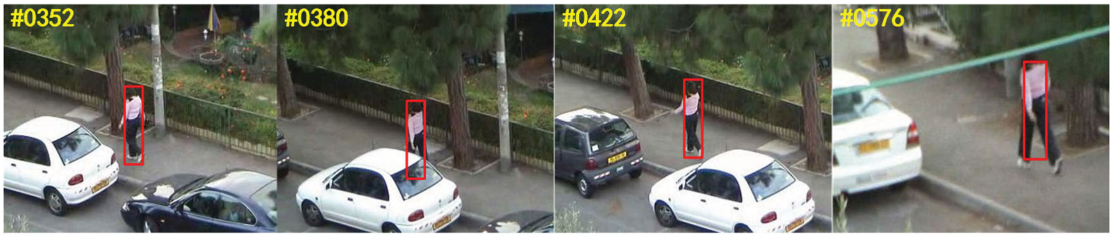

Figure 4. The visualization result of video sequence *Woman*. Our tracker is denoted with a red bounding box.

precision index, we first calculate the Euclidean distance between the estimated and the actual object location in each frame. Then we calculate the ratio of the frame number, whose Euclidean distance is below a given threshold t_p, to the total frame number. Then, the precision can be drawn when the threshold t_p varies from 0 to 50. The success rate is an another index that we consider. The success rate is characterized via the bounding box overlap which is defined as $\eta = |r_e \bigcap r_a|/|r_e \bigcup r_a|$ with r_e and r_a indicating the estimated and the actual bounding box, respectively. Here, the symbol $\bigcap$ and $\bigcup$ denote the intersection and the union of two sets and $|\cdot|$ represents the number of pixels. We compute the ratio of the frame number, whose overlap is above the given threshold t_s, to the total frame number. Then, the succuss rate can be drawn when the threshold t_s varies from 0 to 1. Note that the area under curve (AUC) of the success rates is used to characterize the different trackers.

In this experiment, we consider the following three evaluation methods, i.e. one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). OPE is a traditional evaluation way, where the tracker is tested on a given video sequence with the initial object location and bounding box. In order to eliminate the influence of the sensibility caused by initialization, we change the standard video sequence with some regular means to test the robustness of the tracker. In TRE, we cut each sequence into 20 fragments and we evaluate the tracking algorithm on each subsequence. As for SRE, we shift the given object location and scale the given size of the bounding box in the first frame. There are 8 shift means and 4 scale variations. Subsequently, we evaluate the tracking algorithm on all 12 cases.

Our tracking algorithm and other algorithms are implemented on Benchmark-50 (Wu et al., 2013) by using the OPE, TRE and SRE, and the precision and success rate for all algorithms are given in Figure 3. Note that, in Figure 3, we set the threshold $t_p = 20$ in precision plot and, in success plot, the AUC is adopted. It can be seen from in Figure 3 that our tracking algorithm has high precision and success rate, which indicates that our tracking algorithm is superior over other 10 state-of-the-art tracking methods in terms of the precision and success rate.

In order to display the tracking results of our algorithm, we perform our algorithm on a classical video sequence *Woman* which is subjected to the scale variation and the occlusion. The visualization result is shown in Figure 4. It can be seen from Figure 4 that the occlusion occurs at the 380*th* frame and the scale variation occurs at the 576*th* frame and the proposed tracking algorithm can track the object effectively in the mixed scenario of the scale variation and the occlusion. In our further research, we will consider integrating the proposed algorithm into other tracking models such as the part-based model with the hope of improving further the tracking performance.

6. Conclusions

In this paper, we have proposed a visual object tracking algorithm to handle the problems caused by the scale variation and the occlusion. A KCF has been first learned to acquire the correlation response and the convolutional features extracted from pre-trained VGGNet-19 have been used to strengthen the ability to represent the sample. Then, we have employed the scale adaption strategy to adjust the bounding box during the tracking process. In order to re-find the object location when occlusion occurs, we have proposed a novel re-detection model by using SVM classifier. Extensive experiments have been implemented on the OTB and the results show that the proposed tracking algorithm has a better performance than the other state-of-the-art trackers in terms of the precision and success rate.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported in part by the National Natural Science Foundation of China [grant number 61473076], [grant number 61673103], [grant number 61873059]; the Natural Science Foundation of Shanghai [grant number 18ZR1401500]; the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning of China .

References

Babenko, B., Yang, M. H., & Belongie, S. (2011). Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(8), 1619–1632.

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H. S. (2016, October 8–16). *Fully-convolutional siamese networks for object tracking*. European Conference on Computer Vision, Amsterdam, Netherlands, pp. 850–865.

Bolme, D. S., Beveridge, J. R., Draper, B. A., & Lui, Y. M. (2010, June 13-18). *Visual object tracking using adaptive correlation filters*. IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, pp. 2544–2550.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Danelljan, M., Häger, G., Khan, F., & Felsberg, M. (2014, September 1–5). *Accurate scale estimation for robust visual tracking*. British Machine Vision Conference, Nottingham, UK, pp. 65.1–65.11.

Danelljan, M., Hager, G., Shahbaz Khan, F., & Felsberg, M. (2015, December 13–16). *Convolutional features for correlation filter based visual tracking*. IEEE International Conference on Computer Vision, Santiago, Chile, pp. 58–66.

Danelljan, M., Hager, G., Shahbaz Khan, F., & Felsberg, M. (2015, December 13–16). *Learning spatially regularized correlation filters for visual tracking*. IEEE International Conference on Computer Vision, Santiago, Chile, 4310–4318.

Danelljan, M., Shahbaz Khan, F., & Van de Weijer, J. (2014, June 24–27). *Adaptive color attributes for real-time visual tracking*. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 1090–1097.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Li, F. F. (2009, June 20–25). *Imagenet: A large-scale hierarchical image database*. IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, pp. 248–255.

Girshick, R. (2015, December 13–16). *Fast r-cnn*. IEEE International Conference on Computer Vision, Santiago, Chile, pp. 1440–1448.

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M. M., Hicks, S. L., & Torr, P. H. S. (2016). Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(10), 2096–2109.

Held, D., Thrun, S., & Savarese, S. (2016, October 8–16). *Learning to track at 100 fps with deep regression networks*. European Conference on Computer Vision, Amsterdam, Netherlands, 749–765.

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2012, October 7–13). *Exploiting the circulant structure of tracking-by-detection with kernels*. European Conference on Computer Vision, Firenze, Italy, pp. 702–715.

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 583–596.

Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., & Tao, D. (2015, June 7–12). *Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking*. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 749–758.

Kalal, Z., Matas, J., & Mikolajczyk, K. (2010, June 13–18). *Pn learning: Bootstrapping binary classifiers by structural constraints*. IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, pp. 49–56.

Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(7), 1409–1422.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012, December 3–8). *Imagenet classification with deep convolutional neural networks*. International Conference on Neural Information Processing Systems, Lake Tahoe, USA, pp. 1097–1105.

Li, Y., & Zhu, J. (2014, September 6–12). *A scale adaptive kernel correlation filter tracker with feature integration*. European Conference on Computer Vision, Zurich, Switzerland, pp. 254–265.

Long, J., Shelhamer, E., & Darrell, T. (2015, June 7–12). *Fully convolutional networks for semantic segmentation*. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 3431–3440.

Lucey, S. (2008, June 23–28). *Enforcing non-positive weights for stable support vector tracking*. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, pp. 1–8.

Ma, C., Huang, J. B., Yang, X., & Yang, M. H. (2015, December 13–16). *Hierarchical convolutional features for visual tracking*. IEEE International Conference on Computer Vision, Santiago, Chile, pp. 3074–3082.

Ma, C., Yang, X., Zhang, C., & Yang, M. H. (2015, June 7–12). *Long-term correlation tracking*. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 5388–5396.

Montero, A. S., Lang, J., & Laganire, R. (2015, December 13–16). *Scalable kernel correlation filter with sparse feature integration*. IEEE International Conference on Computer Vision, Santiago, Chile, pp. 587–594.

Nam, H., & Han, B. (2016, June 26–July 1). *Learning multi-domain convolutional neural networks for visual tracking*. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 4293–4302.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015, September 7–10). *Deep face recognition*. British Machine Vision Conference, Swansea, UK, pp. 41.1–41.12.

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., & Yang, M. H. (2016, June 26–July 1). *Hedged deep tracking*. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 4303–4311.

Rifkin, R., Yeo, G., & Poggio, T. (2003). Regularized least-squares classification. *Acta Electronica Sinica*, *190*(1), 93–104.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. Computing Research Repository, abs/1409.1556.

Vedaldi, A., & Lenc, K. (2015, October 26–30). *Matconvnet: Convolutional neural networks for matlab*. ACM International Conference on Multimedia, Brisbane, Australia, pp. 689–692.

Wang, Q., Chen, F., Xu, W., & Yang, M. H. (2015). Object tracking with joint optimization of representation and classification. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(4), 638–650.

Wang, X., Hua, G., & Han, T. X. (2010, September 5–11). *Discriminative tracking by metric learning*. European Conference on Computer Vision, Heraklion, Greece, pp. 200–214.

Wang, M., Liu, Y., & Huang, Z. (2017, July 22–25). *Large margin object tracking with circulant feature maps*. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp. 4021–4029.

Wang, N., & Yeung, D. Y. (2013, December 5–10). *Learning a deep compact image representation for visual tracking*. International Conference on Neural Information Processing Systems, Lake Tahoe, USA, pp. 809–817.

Williams, C. K. I. (2003). Learning with kernels: Support vector machines, regularization, optimization, and beyond. *Publications of the American Statistical Association*, *98*(462), 489–489.

Wu, Y., Lim, J., & Yang, M. H. (2013, June 25–27). *Online object tracking: A benchmark*. IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, pp. 2411–2418.

Zeng, N., Wang, Z., Zhang, H., Liu, W., & Alsaadi, F. E. (2016). Deep belief networks for quantitative analysis of a gold immunochromatographic strip. *Cognitive Computation*, *8*(4), 684–692.

Zeng, N., Zhang, H., Song, B., Liu, W., Li, Y., & Dobaie, A. M. (2017). Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing*, *273*, 643–649.

Zhang, K., Zhang, L., Liu, Q., Zhang, D., & Yang, M. H. (2014, September 6–12). *Fast visual tracking via dense spatio-temporal context learning*. European Conference on Computer Vision. Zurich, Switzerland, pp. 127–141.

Zhang, K., Zhang, L., & Yang, M. H. (2012, October 7–13). *Real-time compressive tracking*. European Conference on Computer Vision, Firenze, Italy, pp. 864–877.

Zhong, W., Lu, H., & Yang, M. H. (2014). Robust object tracking via sparse collaborative appearance model. *IEEE Transactions on Image Processing*, *23*(5), 2356–2368.