RESEARCH ARTICLE

Jointly Feature Learning and Selection for Robust Tracking via a Gating Mechanism

Bineng Zhong*, Jun Zhang, Pengfei Wang, Jixiang Du, Duansheng Chen

Department of Computer Science and Technology, Huaqiao University, Xiamen, Fujian, 361021, China

* bnzhong@hqu.edu.cn

Abstract

To achieve effective visual tracking, a robust feature representation composed of two separate components (i.e., feature learning and selection) for an object is one of the key issues. Typically, a common assumption used in visual tracking is that the raw video sequences are clear, while real-world data is with significant noise and irrelevant patterns. Consequently, the learned features may be not all relevant and noisy. To address this problem, we propose a novel visual tracking method via a point-wise gated convolutional deep network (CPGDN) that jointly performs the feature learning and feature selection in a unified framework. The proposed method performs dynamic feature selection on raw features through a gating mechanism. Therefore, the proposed method can adaptively focus on the task-relevant patterns (i.e., a target object), while ignoring the task-irrelevant patterns (i.e., the surrounding background of a target object). Specifically, inspired by transfer learning, we firstly pre-train an object appearance model offline to learn generic image features and then transfer rich feature hierarchies from an offline pre-trained CPGDN into online tracking. In online tracking, the pre-trained CPGDN model is fine-tuned to adapt to the tracking specific objects. Finally, to alleviate the tracker drifting problem, inspired by an observation that a visual target should be an object rather than not, we combine an edge box-based object proposal method to further improve the tracking accuracy. Extensive evaluation on the widely used CVPR2013 tracking benchmark validates the robustness and effectiveness of the proposed method.

OPEN ACCESS

Citation: Zhong B, Zhang J, Wang P, Du J, Chen D (2016) Jointly Feature Learning and Selection for Robust Tracking via a Gating Mechanism. PLoS ONE 11(8): e0161808. doi:10.1371/journal.pone.0161808

Editor: Quan Zou, Tianjin University, CHINA

Received: June 11, 2016

Accepted: August 14, 2016

Published: August 30, 2016

Data Availability Statement: Data are from the CVPR 2013 tracking benchmark whose authors may be contacted at the following: [1] Y. Wu, J.W. Lim, and M.H. Yang. Online Object Tracking: A Benchmark. IEEE International Conference on Computer Vision and Pattern Recognition, 2013.

Funding: This work is supported by Natural Science Foundation of China (No. 61572205 and 61175121) (http://www.nsfc.gov.cn/), Natural Science Foundation of Fujian Province (No. 2015J01257 and 2013J06014) (http://www.fjkjt.gov.cn/), Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (No.ZQN-PY210 and ZQN-YX108), and 2015 Program for New Century Excellent Talents in

1. Introduction

Visual tracking is a fundamental task in computer vision applications, making it a key component of a real system. Consequently, it has been receiving a huge amount of attention and tremendous progress has been made in visual tracking over the past decades. However, designing robust tracking methods is still an open issue, especially considering various complicated variations that may occur in real-world scenes, e.g., partial occlusion, cluttered backgrounds, illumination changes, motion blur, scale variations, etc.

The performance of a tracking system mainly relies on the used feature representation technique. Typically, the feature representation is composed of two separate components, i.e.,

Fujian Province University. (http://www.hqu.edu.cn/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

feature learning and selection. Towards these two components, a huge number of different methods for visual tracking have been proposed and a variety of features are utilized for modeling an object appearance model. Color or gray feature is widely used in the visual tracking literature to differ a target object from its surrounding backgrounds. In the famous mean shift-based tracking method [1], Comaniciu et al. employ a spatial-weighting color histogram for construing an object appearance model. Instead of using a fixed set of features, Collins et al. [2] propose an online feature ranking-based tracking method for continuously choosing the best set of features used to improve tracking performance. In [3], Possegger et al. propose a discriminative color model-based tracking method via mining distracting regions and adapting an object representation to suppress these regions. Zhang et al. [4] use a LAB color model to extract the features for visual tracking. Recently, Liang et al. [5] present a comprehensive survey on using color information for visual tracking from both the algorithm and benchmark perspectives.

Although color-based tracking methods can provide rich cues to effectively handle partial occlusion and pose variations in visual tracking, they may be sensitive to illumination variations and noises. Therefore, most modern visual tracking methods limit themselves to the more complicated features, e.g., Haar features, histogram of gradients (HoG), local binary pattern (LBP), etc. In addition to using raw pixel values, Henriques et al. [6] use HoG features to further improve tracking performance under a correlation filtering framework using the circulant matrices. Bertinetto et al. [7] propose a correlation filter-based tracking method via combining HoG features and a global color histogram. In [8], Zhang et al. propose a circulant sparse tracker which enables HoG features feasible for sparse representation-based trackers. Grabner and Bischof [9] propose an online adaboost-based tracking method using Haar features. In [10], Avidan propose an ensemble tracking method which uses Haar feature-based weak classifiers to adaptively construct a strong classifier. Takala et al. [11] combine color, LBP and motion features for multi-object tracking. Tong et al. [12] apply LBP features into visual tracking under the tracking-by-detection framework. Some key point-based descriptors are also used for visual tracking, e.g., SFIT and SURF etc. To obtain accurate boundaries of a target object, Fan et al. [13] use SIFT features as a short-term salient points to generate scribbles for robust matting. In [14], a lie algebra-based covariance matrix is utilized for visual tracking. In [15], Wang et al. propose an optimal appearance model-based tracking method, in which multiple cues are effectively integrated in the model. In [16], to effectively deal with multi-modal datasets, an online multi-modal non-negative dictionary learning method is used for visual tracking. However, one major drawback of the above handcrafted feature-based tracking method is that they are incapable to capture semantic information of targets, and not robust to significant appearance changes. On the other hand, the separated feature learning and selection component easily lead to the learned features not all relevant and noisy.

Recently, inspired by the success of deep learning in a variety of computer vision tasks [17–19], a large amount of deep leaning-based tracking methods have been proposed [20–28] for improve tracking performance. In [20], Fan et al. propose a convolutional neural network-based human tracking method which pre-learns the human-specific features during offline training. Wang and Yeung [21] propose a two-layer auto-encoder based tracker which is firstly pre-trained offline and then fine-tuned for an online tracking task. However, the discriminative power of the learned deep features may be limited due to the pre-training is performed in an unsupervised way. In [22], multiple convolutional neural networks are used for visual tracking. To further improve the discriminative power, some authors pre-train deep convolution networks on a large-scale image classification task (i.e., Imagenet) and then fine-tuned for a specific tracking task. By simultaneously using feature maps of multiple convolution layers from the VGG, Wang et al. [23] propose a fully convolutional neural network-based tracking method. In [24], Hong et al. employ a convolutional neural network which is pre-trained on Imagenet to predict saliency maps for online tracking. Ma et al. [25] firstly exploit feature maps from multiple convolution layers of a deep VGG to train multiple correlation filters. Then, the foreground heat maps estimated by the correlation filters are combined to provide robust tracking results. In [26], a multi-domain CNNs, composed of shared layers and multiple branches of domain-specific layers, is trained using a large set of videos with tracking ground truths for visual tracking. Each domain is trained for individual videos and each branch is used to classify a target object in each domain. In [27], CNN-based tracking method is proposed, in which a Hedge method is used to combine several CNN trackers from different CNN layers into a stronger one. To effectively transfer pre-trained deep features for online tracking, Wang et al. [28] present a sequential training method for convolutional neural networks. In [29], Tao et al. use a Siamese network for visual tracking. The Siamese network is pre-trained in a large and external videos to learn a matching mechanism. Despite achieving state-of-the-art tracking performance in recent benchmark evaluations [30, 31], most existing deep learning-based tracking methods still have some limitations due to blindly learn a representation using the majority of the learned high-level features.

In addition to focus on the feature representation, some authors have modeled an object appearance model using numerous advanced classifiers. The typical classifiers include correlation filters [6, 32, 33], ensemble learning [9, 34, 35], support vector machine [36–38], P-N learning [39], random forests [40], multiple instance learning [41], metric learning [42, 43], and sparse coding and low-rank matrices [44, 45] etc.

Recently, object proposal has made much progress for object detection [46–49] and segmentation. Inspired by this, several object proposal-based approaches [50–52] have also been proposed for robust visual tracking. In [50], visual tracking is viewed as an object proposal selection task. A fusion of detection confidence score, edges and motion boundaries is used to locate a target object. In [51], BING-based object proposal algorithm is adopted for visual tracking. To reduce a large amount of test space and provide a better training set for a tracker, Zhu et al. [52] employ an edge-ness based object proposal method for visual tracking. For a more comprehensive reviews on visual tracking methods, please refer to ([30, 31, 53] and [54]).

Despite achieving state-of-the-art tracking performance, most of the above visual tracking methods share a same basic assumption that the raw video sequences are clear. This assumption, however, may be too restrictive, especially under difficult conditions such as a complex real-world scene with significant noise and irrelevant patterns. In other words, most of the above tracking methods may fail if there is no good raw features to start with.

In this paper, to address the above-mentioned issues, we propose a novel unsupervised tracking algorithm via a point-wise gated convolution deep network (CPGDN) [55] that combines feature learning and feature selection coherently in a unified framework. Specially, the CPGDN is firstly pre-trained to automatically learn and select partially useful high-level abstractions from extracted image features on a Tiny image dataset [56]. Secondly, the CPGDN is further fine-tuned to adapt to a specific target object during online tracking. The proposed CPGDB-based tracker performs dynamic feature selection from the raw videos when the task-relevant patterns occur through a gating mechanism. Intuitively speaking, the model can adaptively focus on a variable subset of visible nodes corresponding to a specific target object instead of its surrounding backgrounds. Finally, to further improve tracking performance, we effectively incorporate an object proposal-based method (i.e., edge box-based proposals [46]) into the CPGDN-based tracker. This is inspired by an observation that most trackers are easily prone to locate on a non-object target (i.e., a background object or textureless object) when the trackers have failed. Obviously, if a target object is non-object, the edge response is weak and the edge score is near zero. Therefore, we use an edge box-based proposal scoring function as a complementary cue to adjust the tracking results. We make an edge box based proposal score be negative if the edge box-based proposal method detects the non-object. A simple yet effective fusion schema is designed to combine the CPGDN model based score and the edge box-based proposal score. Extensive experiments on the CVPR2013 tracking benchmark [30], containing 50 sequences and 29 publicly available trackers, validate the robustness and effectiveness of the proposed tracking method. The main contributions of this work are three folds.

1. First, we design a unified feature learning and selection framework for visual tracking, in which the proposed tracking method is equipped with a CPGDN model trained end-to-end on the Tiny image dataset [56]. Consequently, the proposed tracking method is robust to the object appearance variations in video sequences.
2. Second, on the basis of the learnt object appearance model using the CPGDN model, we incorporate an edge box-based proposal scoring function into the object appearance model to further improve tracking performance.
3. Third, extensive experiments in the CVPR 2013 tracking benchmark [30] show that the proposed CPGDN-based tracker can achieve promising performance compared to the state-of-the-art trackers.

The rest of the paper is organized as follows. In Section 2, the proposed CPGDN-based tracking method is described in details. Then, we present an extensive evaluation of the

Table 1. Overview of the proposed CPGDN-based tracking method.

Algorithm 1 Jointly Feature Learning and selection for Robust Tracking via a Gating Mechanism	
Input:	
	1. Pre-trained CPGDN filters $\{w^1,w^2,w^3\}$
	2. Initial target state x_1.
Output:	
	1. Estimated target states x_t^*.
Initialization:	
	1. Initialize particles.
	2. Randomly initialize the last full connect layer w^4.
	3. Collect positive samples s_1^+ and negative samples s_1^-.
	4. Construct the CPGDN-based appearance model via fine-tuning using s_1^+ and s_1^-.
for t = 2 to the end of the video	
1. Prediction: apply a prediction function in a particle filtering framework to obtain a set of candidate samples/particles $\{c_i\}_{i=1}^N$	
2. Likelihood evaluation:	
	(1) Calculate a detection score $f_T(c_i)$ based on the CPGDN model for each particle $\{c_i\}_{i=1}^N$.
	(2) Calculate an edge box-based score $f_E(c_i)$ using an edge box-based object proposal method for each particle $\{c_i\}_{i=1}^N$.
	(3) Find the optimal target state x_t^* by Eq (6).
3. Model updating:	
	(1) Generate new positive and negative samples s_t^+ and s_t^- according to the optimal target state.
	(2) Update CPGDN-based appearance model using new positive samples s_t^+ and new negative samples s_t^- if the score of the optimal target state below a threshold φ.
end for	

doi:10.1371/journal.pone.0161808.t001

proposed CPGDN-based tracker and demonstrate the experimental results in Section 3. Finally, we conclude remarks in Section 4.

2. The Proposed CPGDN-Based Tracking Method

In this section, we present our tracking method via a point-wise gated convolutional deep network (CPGDN), which can jointly performs feature learning and selection in a unified framework. Table 1 schematically show the proposed CPGDN-based tracking method under a particle filtering framework.

Specifically, the main components of the proposed CPGDN-based tracking method are: (i) In an initial frame, we firstly collect some positive samples and negative samples, where positive and negative examples have more than 0.7 and less than 0.5 the Intersection over Union (IoU) overlap ratios with ground-truth bounding boxes. Then, the CPGDN model pre-trained on a large-scale image data set (i.e., Tiny image dataset [56]) is fine-tuned according the positive and negative samples. (ii) In subsequent frames, a set of candidate samples are firstly generated by a prediction function within a particle filtering framework. Then, the final scores for each candidate sample is determined by fusing both scores from the CPGDN model and the edge box-based proposal method. (iii) The optimal target location is determined by the candidate sample with the maximum score. (iv) The CPGDN model is updated if the maximum score of candidate samples below a threshold φ. The tracking procedure continues in this iterative fashion until the end of video. Each detained component of the proposed CPGDN-based tracking method is described in the following subsections.

2.1 Visual tracking under a particle filtering framework

The proposed CPGDN-based tracking method is carried out using a particle filtering framework [57] which is a technique for implementing recursive Bayesian filter by Monte Carlo sampling. The key idea is to represent the posterior density by a set of random particles/samples with associated weights. The posterior probability can be estimated based on these samples and weights.

Suppose we have an observation of a target object $Y_t = \{y_1,\ldots,y_t\}$ up to the t^{th} frame, the posterior probability $p(x_t \mid Y_t)$ can be calculated by the Bayesian theorem as the following:

$$p(x_t|Y_t) \propto p(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|Y_{t-1}) dx_{t-1} \tag{1}$$

where $p(x_t \mid x_{t-1})$ is a prediction function, and $p(y_t \mid x_t)$ is a likelihood evaluation function which determines the likelihood of observing y_t at state x_t. The optimal object state x_t^* at time t can be inferred as follows

$$x_t^* = \arg\max_{x_t^i}^{i=1,\ldots,\mathrm{N}} \{p(y_t^i|x_t^i) p(x_t^i|x_{t-1})\} \tag{2}$$

where x_t^i is the i^{th} sample of the state x_t, and y_t^i is the image observation predicted by x_t^i. In this paper, a target state is denoted by $x_t = (\mathrm{l}_t^x, \mathrm{l}_t^y, \mathrm{w}_t, \mathrm{h}_t)$ where the four parameters are the horizontal coordinate, vertical coordinate, width and height respectively. The prediction function $p(x_t \mid x_{t-1})$ is modeled by a Normal distribution function, i.e., $p(x_t \mid x_{t-1}) = N(x_t; x_{t-1}, \Sigma)$, where Σ is a diagonal covariance matrix whose diagonal elements are the corresponding variances of respective parameters. In order to estimate likelihood of each state x_t, we firstly normalize each image patch (i.e., each particle sample) to $32*32$ pixels. Then, the likelihood of each particle is calculated based on the CPGDN model, i.e., $p(y_t \mid x_t) = d_t$, where d_t is an output score estimated from the CPGDN model.

2.2 The CPGDN based appearance model

In this section, we address the problem of how to learn a CPGDN based appearance model via jointly feature learning and selection in a unified framework. We construct a two-layer CPGDN model, in which the first layer is composed by convolutional restricted Boltzmann machines (CRBM) and the second layer is composed by convolutional point-wise gated Boltzmanne machine (CPGBM) followed by a full connection layer.

More specifically, we use the proposed CPGDN [55] to extract features of a target object. The key advantages of CPGDN is convolutional architecture and jointly performing the feature learning and selection in a unified framework. Convolutional architecture is good at dealing with spatially correlated data while feature selection can obtain more robust features from complex real-word data. Inspired by these advantages from the CPGDN model, in this paper, we propose a CPGDN-based method to effectively learn the abstract feature to distinguish a target object from the non-target objects.

The CPGDN model is illustrated in Fig 1. Following the notations of Sohn et al. [55], we will briefly review the CPGDN model and focus on how to construct the CPGDN model based appearance model for visual tracking.

The generic feature extraction based on CRBM. We use the convolutional RBM with probabilistic max pooling (CRBM) to extract the generic features. Please see [58] for more details about CRBM. The CRBM is composed by a "detection" layer, which is similarly to the convolutional layer of CNN, and a "pooling" layer, which shrink the representation of the detection layer. The CRBM with pooling layer are more robust to small variations. Denote $\mathrm{I} \in \mathbb{R}^{N \times N \times C}$ as the input image, where C denotes the number of input channels (e.g., $C = 1$ for gray images) and K denotes the number of filters. $ws \times ws$ as the 2D convolutional filter size. W

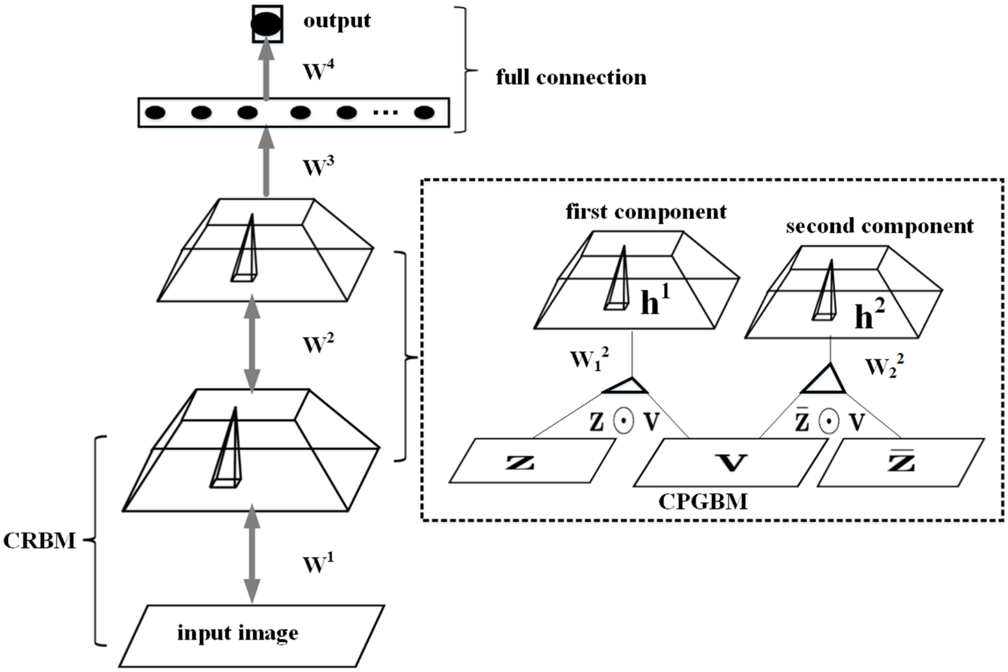

Fig 1. Architecture of the two-layer CPGDN model [55] with a full connection layer. The first layer is CRBM, and the second layer is CPGBM with two mixture components. z is a gating mechanism and its value is binary variable. z and $\bar{z}$ are complementary, i.e., $\bar{z} = 1 - z$. We use the first component of CPGBM as the input of a full connection layer.

doi:10.1371/journal.pone.0161808.g001

denotes the square filters of size s, i.e., $\mathrm{W}^{k,c} \in \mathbb{R}^{ws \times ws}$. The operator $\tilde{\mathrm{W}}$ applied in the matrix W denotes the vertical and horizontal flip of the matrix. Please see the experimental section for more details on parameter setting.

The semantic feature learning and selection based on CPGBM. Once trained the CRBM, we use the output of pooling layer in CRBM as the input of the CPGBM. Denote $\mathrm{z}_{m,n}$ as the switch units. Denote R as the mixture components. z have the same size with v. Note that all the channels of input v shared the same switch unites z. Intuitively speaking, every channel of input sample shared the same switch unites. Given the other two types of variables, we can compute the conditional probabilities of hidden, switch, and visible units below:

$$P(h^{r,k}_{i,j} = 1|\mathrm{v}, \mathrm{z}) = \sigma\Big(\sum_c (\tilde{\mathrm{W}}^{r,k,c} * (\mathrm{z}^r \odot \mathrm{v}^c))_{i,j} + b^r_k\Big) \tag{3}$$

$$P(z^r_{m,n} = 1|\mathrm{v}, \mathrm{h}) = \frac{\exp\,(v^c_{m,n}(\sum_k (\mathrm{W}^{r,k,c*}\mathrm{h}^{r,k})_{m,n} + c^r_c))}{\sum_s \exp\,(v^c_{m,n}(\sum_k (\mathrm{W}^{s,k,c*}\mathrm{h}^{s,k})_{m,n} + c^s_c))} \tag{4}$$

$$P(v^c_{m,n} = 1|\mathrm{h}, \mathrm{z}) = \sigma\Big(\sum_r z^r_{m,n} \Big[\sum_k (\mathrm{W}^{r,k,c*}\mathrm{h}^{r,k})_{m,n} + c^r_c\Big]\Big) \tag{5}$$

The operator e denotes an element-wise multiplication between two matrices. The above equations subject to $\sum_{r=1}^{R} z^r_{m,n} = 1$. Please note that the CPGBM only has convolutional layer and we use the first components of CPGBM as the learnt features.

Learning object appearance models from CPGDN. The CPGDN is composed by stacking the CPGBM on the first layer of CRBM. This construction makes sense because the first layer mostly learn the generic features and the higher layer learn semantic features. But, not all semantic features are good for our task and we need typical semantic features for our specific target object. Firstly, we use a large number of images from the Tiny image dataset [56] to offline train the CPGDN model with one fully connection layer. Then we transfer the learned parameters to initial the model used for online tracking. Typically, we can get a ground-truth bounding box of a target object in an initial frame. A warping technology is utilized to generate the positive and negative samples. The positive and negative examples have more than 0.7 and less than 0.5 IoU overlap ratios with the ground-truth bounding boxes. The generated positive and negative examples are used to fine-tune the pre-trained CPGDN model. During the tracking process, we update the CPGDN model using the newly observed target samples when the maximum confidence of all particles/samples is below a pre-defined threshold φ.

2.3 An edge box-based object proposal method

In this paper, we employ an efficient edge box-based object proposal method [46] for further improving tracking results. The goal is to make our tracker focus on a visual target object.

Specifically, based on a key idea that a bounding box likely contains a visual object if the number of contours wholly enclosed by the bounding box is enough, the edge box-based object proposal method generates a set of object candidates. Firstly, a structured forest based edge detector [59] is used to estimate an edge map for each pixel in an input image. Then, given the extracted edge map, a pool of sampled bounding boxes is generated via a sliding window way. Finally, according to the number of contours wholly enclosed by a bounding box, the score for a bounding box is calculated. For more details, please refer to [46].

2.4 The CPGDN-based tracker driven by edge box-based object proposals

In this section, we utilize the edge box-based object proposal method to improve the performance of the proposed CPGDN-based tracker while maintaining its required computational efficiency.

Without loss of generality, suppose we have a set of candidate particles/samples $\{c_i\}_{i=1}^{N}$ at the t^{th} frame. Based on the CPGDN model and the edge box-based object proposal method, we evaluate the likelihood of a candidate sample c_i belonging to the target object. Firstly, we calculate the CPGDN model based score $f_T(c_i)$ of the candidate sample c_i. Then, we calculate the object proposal score $f_E(c_i)$ of the candidate sample c_i via the edge box-based object proposal method. Consequently, the final score for the candidate sample c_i is calculated as follows.

$$f(c_i) = f_T(c_i) + (f_E(c_i) + \lambda) \tag{6}$$

The value of parameter λ depend on the value of $f_E(c_i)$ and is calculated as follows.

$$\lambda = \begin{cases} 0, & \mathit{if}\ f_E(c_i) > 0 \\ -0.1, & \mathit{else}, \end{cases} \tag{7}$$

We use the simple yet effective fusing schema to combine the CPGDN-based appearance model with the edge box-based object proposal method. The goal is to make the proposed CPGDN-based tracker focus on a visual target object instead of the non-targets due to the edge box-based object proposals can provide rich information for the proposed CPGDN-based tracker. Consequently, the performance of the proposed CPGDN-based tracker driven by edge box-based object proposals can be greatly improved.

3. Experiments

In this section, we introduce extensive experimental results from the proposed CPGDN-based tracker (named CPGDN). Firstly, we describe the setting of our experiments including the implementation details and the evaluation protocol of the CVPR 2013 tracking benchmark [30]. Then, we compare the proposed CPGDN-based tracker with the state-of-the-art trackers on the CVPR 2013 tracking benchmark. Moreover, to verify the effectiveness of the edge box based object proposals method, we compare the standard CPGDN-based tracker with its variant without using the edge box based object proposals method. Finally, we discuss some issues and future work.

3.1 Experiment setting

We implement the proposed CPGDN-based tracker in MATLAB. The running speed is about one frame per second on a HP Z800 workstation with an Intel i5-3470 3.20GHz CPU and 22GB RAM. The number of particles are $N = 600$. For feature extraction, each image patch of a target object is warped to $32*32$ pixels. In the first layer of CPGDN, we set $ws = 5$ and $K = 12$. To get positive and negative examples, we firstly use a warping technology to the target sample and obtain $N_1 = 10$ positive samples. Then, we extracted $N_2 = 100$ negative samples surrounding the target region. The positive and negative examples have more than 0.7 and less than 0.5 IoU overlap ratios with the ground-truth bounding boxes. In the tracking process, once the maximum confidence of all particles/samples in a frame is below a predefined threshold $\varphi = 0.8$, we update the CPGDN model using the new observed target samples. The same parameters are fixed for all of the experiments.

To extensive evaluate the proposed CPGDN-based tracker, we adopt the widely used one-pass evaluation (OPE) metric from the CVPR 2013 tracking benchmark [30] which contains 50 fully annotated image sequences. The 50 image sequences is tagged by 11 tracking challenging factors, such as illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter and low resolution. Experimental results are reported using the precision plots and success plots, which rank trackers in terms of center location error at threshold 20 pixels and area under the curve, respectively. Initially, three are 29 trackers are adopted in the benchmark. For more details, please refer to the paper [30]

3.2 Comparison with other trackers

Quantitative Evaluation. In Fig 2, we show the OPE evaluation results with 29 state-of-art trackers on 50 image sequences [30], where only the top 10 trackers are shown for clarity. In addition, for fair comparisons, we also compare the four recent representative trackers including MEEM [4], KCF [6], DLT [21], and TGPR [60]. Fig 2 shows that the proposed CPGDN-based tracker performs favorably against the state-of-the-art methods on the OPE evaluation metric. More specifically, the proposed CPGDN-based tracker ranks 4th in terms of the precision rate while 3rd based on the success rate. It outperforms DLT by 11.8% in the precision plot and 8.6% in the success plot respectively. In terms of the success plot, the proposed CPGDN-based tracker outperforms KCF. Please note that the key advantage of the proposed CPGDN-based tracker is that it can jointly perform feature selection on raw features through a gating mechanism. Therefore, the proposed CPGDN-based tracker can adaptively focus on the task-relevant patterns (i.e., a target object), while ignoring the task-irrelevant patterns (i.e., the surrounding background of a target object).

Attribute-based Evaluation. To thoroughly evaluate the performance of the proposed CPGDN-based tracker in various challenging scenes, we summarize the performance based on 11 different factors on 50 image sequences [30]. Due to space limitation, we only show the success plots for eight challenge attributes in Fig 3. As shown in Fig 3, the proposed CPGDN-based tracker performs well against the other methods in almost all tracking attributes.

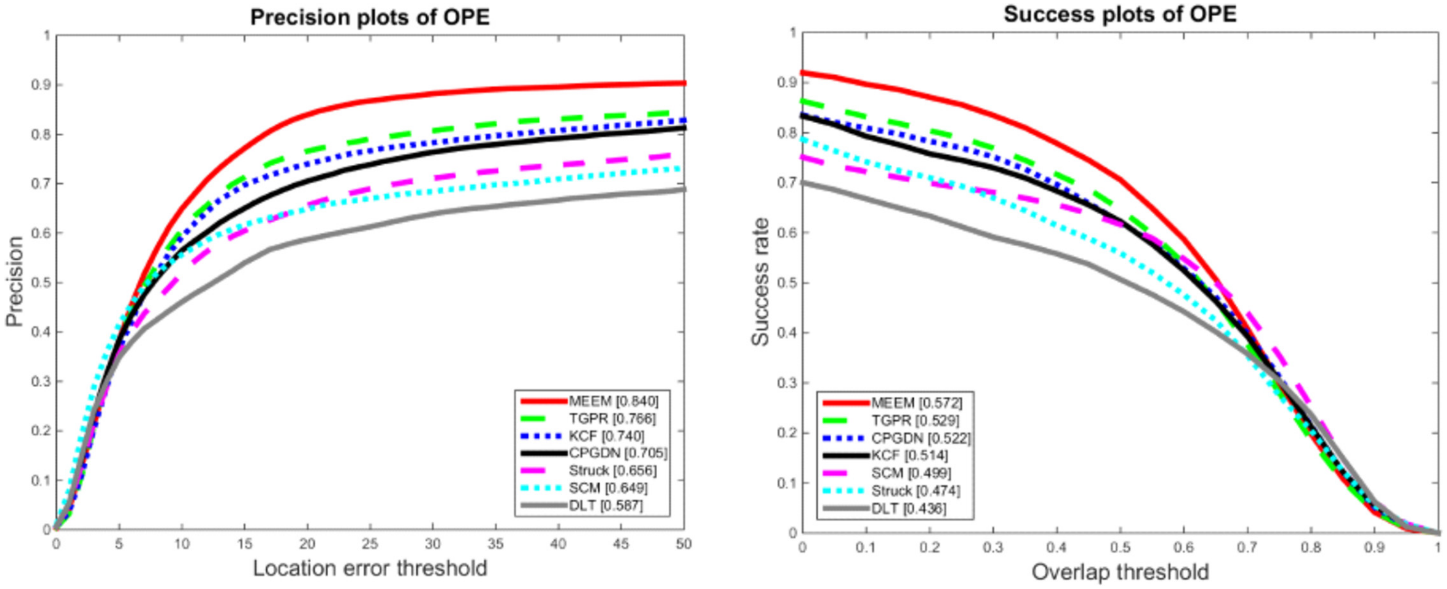

Fig 2. The precision and success plots of quantitative comparison for the 50 sequences in the CVPR2013 tracking benchmark [30].The performance score of each tracker is shown in the legend. The proposed CPGDN-based tracker (named CPGDN) ranks 4th in precision plots and 3th in success plots respectively.

doi:10.1371/journal.pone.0161808.g002

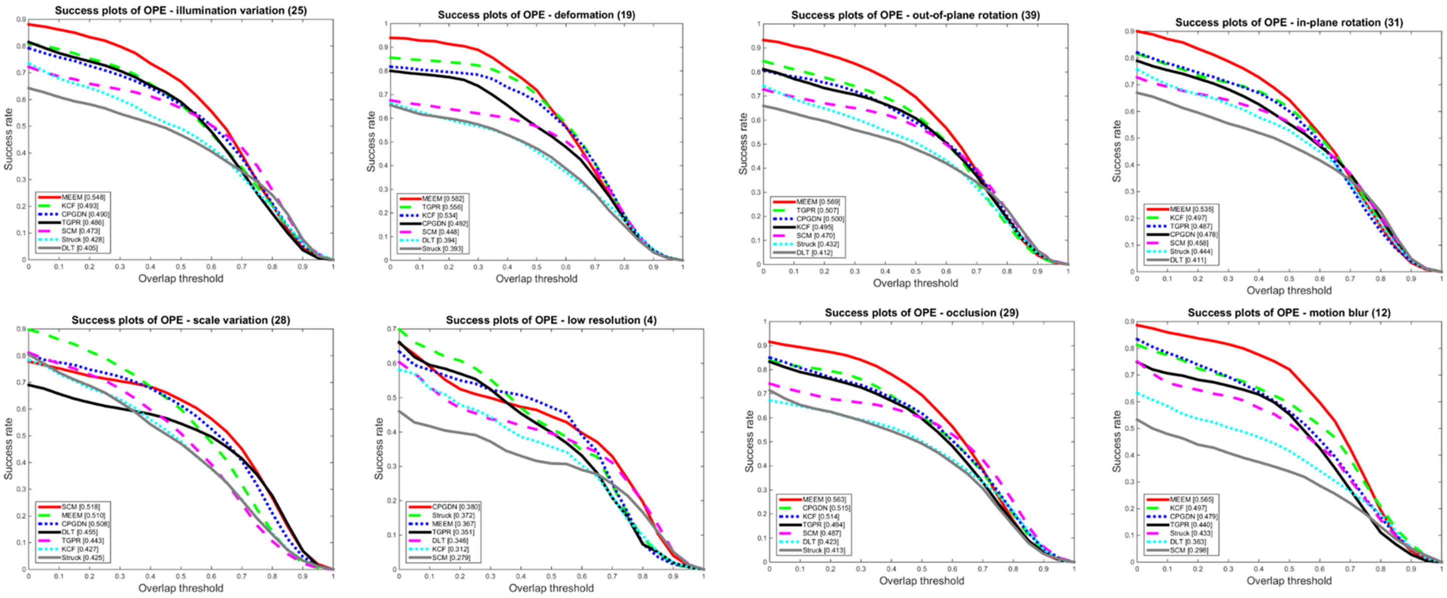

Fig 3. The success plots for eight challenge attributes: illumination variation, deformation, out-of-plane rotation, in-plane rotation, scale variation, low resolution, occlusion, motion blur.

doi:10.1371/journal.pone.0161808.g003

Center Distance Error Evaluation. In Fig 4, we show the center distance error per frame for the four typical image sequences, i.e., the singer2, deer, walking2, and freeman1 sequence respectively. For presentation clarity, the results by the top 8 trackers are shown. The proposed CPGDN-based tracker can achieve promising results due to jointly learning and selecting robust features via the CPGDN model driven by the edge box based object proposals method.

3.3 Efficacy of the edge box based object proposals method

To verify the effectiveness of proposed edge box based object proposals method for the proposed CPGDN-based tracker, we evaluate the performance of the proposed CPGDN-based tracker without using the edge box based object proposals method. Fig 5 shows the quantitative results on the CVPR 2013 tracking benchmark [30]. As shown in Fig 5, without the edge box based object proposals method, both the precision and success rate reduce to some extent. The precision and success rate reduce about 4.3% and 2.6% respectively. This is consistent to our intuition that a tracker should focus on an object target instead of a non-object target. By combing \the edge box based object proposals method with the CPGDN model, we can effectively alleviate tracker drifting problem to some extent.

3.4 Discussion

Although the proposed CPGDN-based tracker has achieved promising results compared with the state-of-the-art trackers, its performance is a bit worse than those of MEEM, TGPR and KCF. In other words, it is still far from perfect. Here we analyze some causes leading to the failure and discuss some possible solutions:

1. The proposed CPGDN-based tracker transfers generic image features that are more robust against variations from pre-training to online tracking. However, due to the powerful invariant feature representation of the CPGDN model, the proposed CPGDN-based tracker

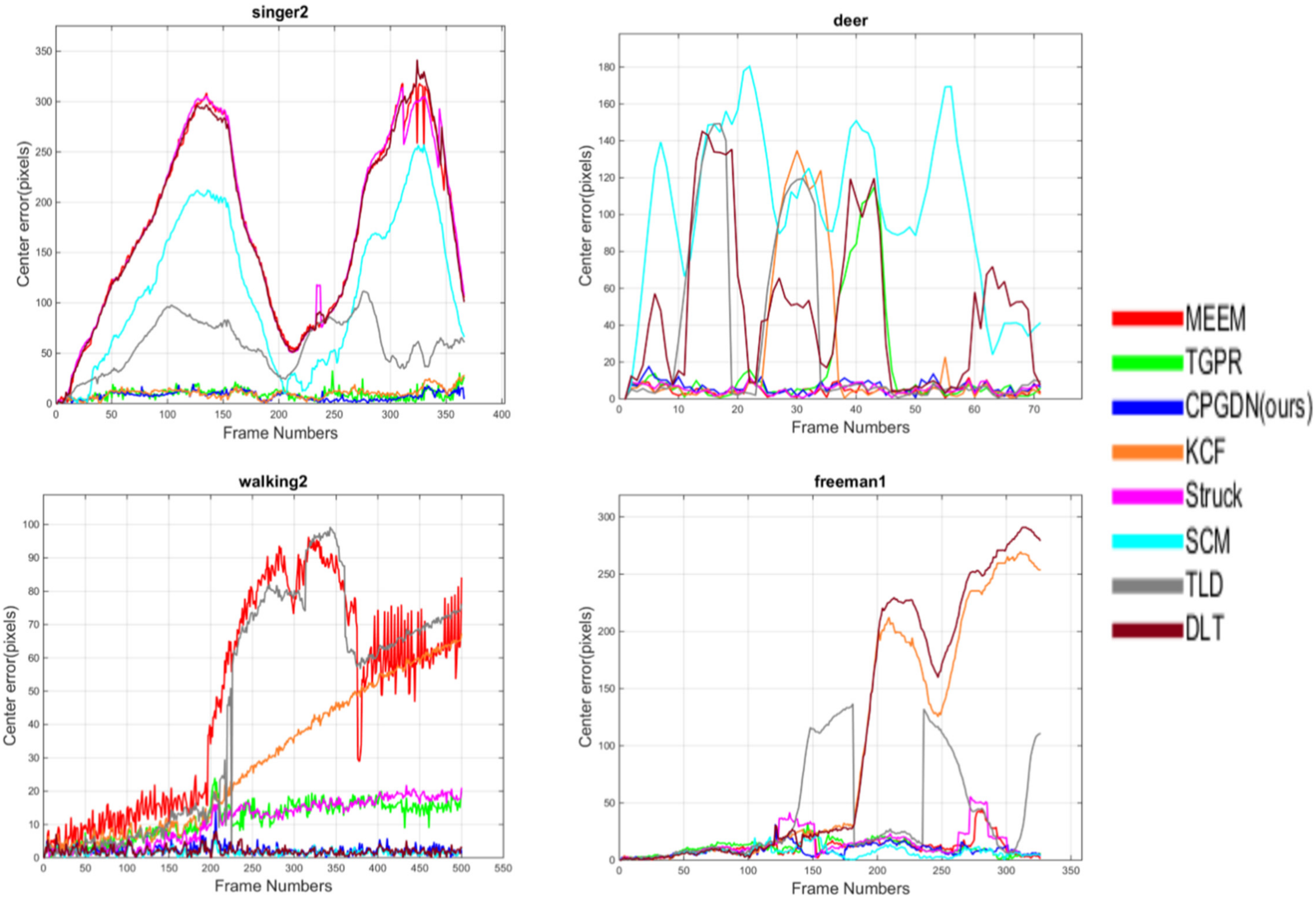

Fig 4. Quantitative comparison on the center distance error per frame for the four image sequences from [30].

doi:10.1371/journal.pone.0161808.g004

may possibly drift when tracking a specific target object which has similar appearance with a distractor.

2. The proposed CPGDN-based tracker is likely to drift when the appearance variations of a target object is huge.

There may be two possible methods to solve the above mentioned issues:

1. In addition to solely relying on the pre-trained CPGDN model, we could build another online appearance model. The online learning-based appearance model can capture the latest appearance variations. These two models can be co-trained to decide the best target state.
2. More effective update strategies could be adopted to improve the tracking results due to an good update strategy can avoid bad samples corrupting the appearance model.

4. Conclusion

In this paper, instead of directly using learned features which probably have some noise for tracking, we have proposed a novel visual tracking method via a point-wise gated convolutional deep network (CPGDN) that jointly performs the feature learning and feature selection in a

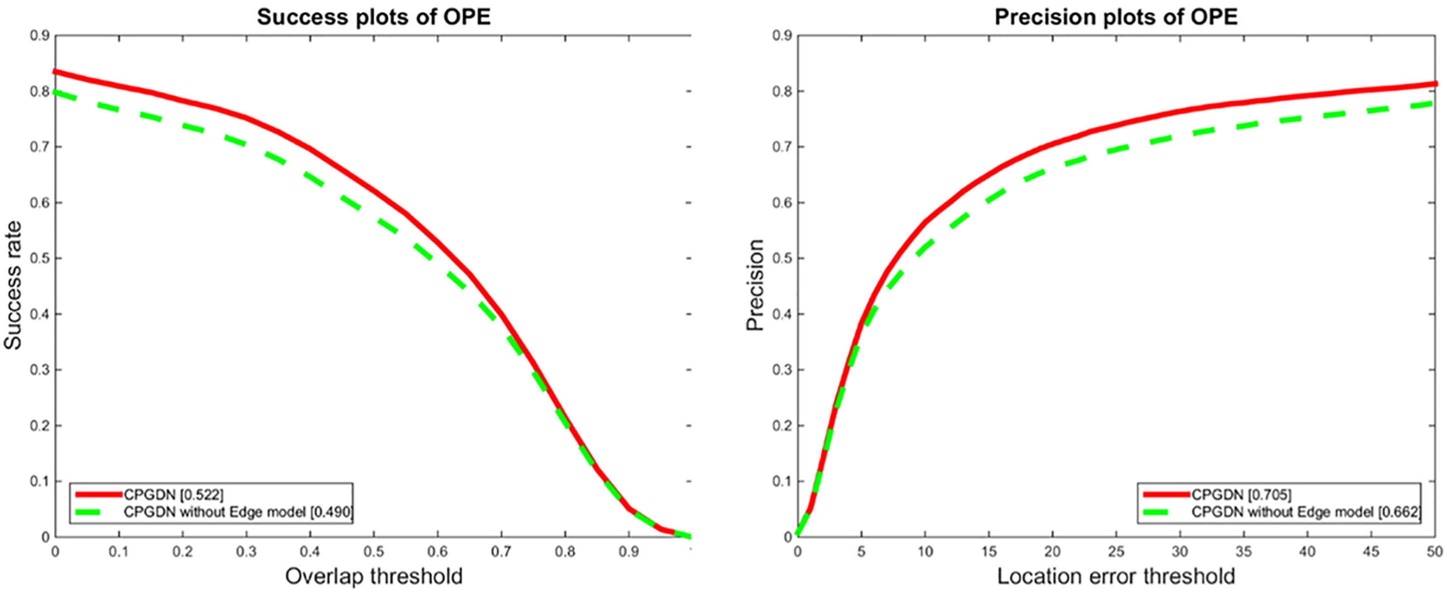

Fig 5. The success plots and precision plots of OPE for the standard CPGDN-based tracker and its variant without the edge box based object proposals method. It is obvious that the proposed CPGDN-based tracker driven by the edge box based object proposal method can achieve promising tracking results.

doi:10.1371/journal.pone.0161808.g005

unified framework. Moreover, our model can adaptively select learned features aiming at different target objects. Based on the observation that a visual target should be an object rather than not, we combine an edge box-based object proposal method with the CPGDN based model to effectively alleviate the tracker drifting problem. Extensive experiments on the CVPR2013 tracking benchmark have validated the robustness and effectiveness of the proposed CPGDN-based tracker.

Acknowledgments

This work is supported by Natural Science Foundation of China (No. 61572205 and 61175121), Natural Science Foundation of Fujian Province (No. 2015J01257 and 2013J06014), Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (No.ZQN-PY210 and ZQN-YX108), and 2015 Program for New Century Excellent Talents in Fujian Province University.

Author Contributions

Conceptualization: BNZ JZ PFW.

Data curation: BNZ JXD DSC.

Formal analysis: BNZ JXD DSC.

Funding acquisition: BNZ JXD.

Investigation: BNZ JZ PFW.

Methodology: BNZ JZ PFW.

Project administration: BNZ.

Resources: BNZ.

Software: JZ PFW.

Supervision: BNZ JXD DSC.

Validation: BNZ JZ PFW.

Visualization: BNZ JXD DSC.

Writing – original draft: BNZ JZ PFW.

Writing – review & editing: BNZ JZ PFW.

References

1. Comaniciu D, Ramesh V, Meer P. Kernel-Based Object Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564–577, 2003.
2. Collins RT, Liu Y, Leordeanu M. Online Selection of Discriminative Tracking Features. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1631–1643, 2005. PMID: 16237997
3. Possegger H, Mauthner T, Bischof H. In Defense of Color-based Model-free Tracking. IEEE International Conference on Computer Vision and Pattern Recognition, 2015.
4. Zhang J, Ma S, Sclaroff S. MEEM: Robust Tracking via Multiple experts using entropy. European Conference on Computer Vision, 2014.
5. Liang P, Blasch E, Ling H. Encoding Color Information for Visual Tracking: Algorithms and Benchmark. IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 5630–5644, 2015. doi: 10.1109/TIP.2015.2482905 PMID: 26415202
6. Henriques JF, Caseiro R, Martins P, Batista J. High-speed Tracking with Kernelized Correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583–596, 2015. doi: 10.1109/TPAMI.2014.2345390 PMID: 26353263
7. Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr PHS. Staple: Complementary Learners for Real-Time Tracking. IEEE International Conference on Computer Vision and Pattern Recognition, 2016.
8. Zhang TZ, Bibi A, Ghanem B. In Defense of Sparse Tracking: Circulant Sparse Tracker. IEEE International Conference on Computer Vision and Pattern Recognition, 2016.
9. Grabner H, Bischof H. On-line Boosting and Vision. IEEE International Conference on Computer Vision and Pattern Recognition, 2006.
10. Avidan S. Ensemble Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 2, pp. 261–271, 2007. PMID: 17170479
11. Takala V, Pietikainen M. Multi-Object Tracking Using Color, Texture and Motion. IEEE Workshop on Visual Surveillance, 2007.
12. Tong ML, Han H, Lei JS. Efficient Visual Tracking by Using LBP Descriptor. Artificial Intelligence and Computational Intelligence, vol. 530, pp. 391–399.
13. Fan JL, Shen XH, Wu Y. Scribble Tracker: A Matting-based Approach for Robust Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 8, pp. 1633–1644, 2012. doi: 10.1109/TPAMI.2011.257 PMID: 22745003
14. Porikli F, Tuzel O, Meer P. Covariance Tracking using Model Update based on Lie Algebra, IEEE International Conference on Computer Vision and Pattern Recognition, 2006.
15. Wang YR, Jiang LK, Liu QY, Yin MH. Optimal Appearance Model for Visual Tracking. PLoS ONE 11 (1): e0146763. doi: 10.1371/journal.pone.0146763 PMID: 26789639
16. Zhang X, Guan NY, Tao DC, Qiu XG, Luo ZG. Online Multi-Modal Robust Non-Negative Dictionary Learning for Visual Tracking. PLoS ONE 10(5): e0124685. doi: 10.1371/journal.pone.0124685 PMID: 25961715
17. Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing System, 2012.
18. Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013. doi: 10.1109/TPAMI.2013.50 PMID: 23787338
19. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-scale Image Recognition. CoRR, abs/1409.1556, 2014.

20. Fan JL, Xu W, Wu Y, Gong YH. Human Tracking using Convolutional Neural Networks. IEEE Transactions on Neural Networks, vol. 21, no. 10, pp. 1610–1623, 2010. doi: 10.1109/TNN.2010.2066286 PMID: 20805052

21. Wang NY, Yeung DY. Learning a Deep Compact Image Representation for Visual Tracking. Neural Information Processing System, 2013.

22. Li HX, Li Y, Porikli F. Deeptrack: Learning Discriminative Feature Representations by Convolutional Neural Networks for Visual Tracking. British Machine Vision Conference, 2014.

23. Wang LJ, Ouyang WL, Wang XG, Lu HC. Visual Tracking with Fully Convolutional Networks. IEEE International Conference on Computer Vision, 2015.

24. Hong S, You T, Kwak S, Han B. Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network. International Conference on Machine Learning, 2015.

25. Ma C, Huang JB, Yang XK, Yang MH. Hierarchical Convolutional Features for Visual Tracking. IEEE International Conference on Computer Vision, 2015.

26. Nam HS, Han BY. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. arXiv:1510.07945, 2015.

27. Qi YK, Zhang SP, Qin L, Yao HX, Huang QM, Lim JW, et al. Hedged Deep Tracking. IEEE International Conference on Computer Vision and Pattern Recognition, 2016.

28. Wang LJ, Ouyang WL, Wang XG, Lu HC. STCT: Sequentially Training Convolutional Networks for Visual Tracking. IEEE International Conference on Computer Vision and Pattern Recognition, 2016.

29. Tao R, Gavves E, Smeulders AWM. Siamese Instance Search for Tracking. IEEE International Conference on Computer Vision and Pattern Recognition, 2016.

30. Wu Y, Lim JW, Yang MH. Online Object Tracking: A Benchmark. IEEE International Conference on Computer Vision and Pattern Recognition, 2013.

31. Smeulders AWM, Chu DM, Cucchiara R, Calderara S, Dehghan A, Shah M. Visual Tracking: an Experimental Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 7, pp. 1442–1468, 2014. doi: 10.1109/TPAMI.2013.230 PMID: 26353314

32. Tang M, Feng JY. Multi-kernel Correlation Filter for Visual Tracking. IEEE International Conference on Computer Vision, 2015.

33. Chen Z, Hong ZB, Tao DC. An Experimental Survey on Correlation Filter-based Tracking. arXiv:1509.05520, 2015.

34. Wen L, Cai Z, Lei Z, Li S. Online Spatio-temporal Structural Context Learning for Visual Tracking. European Conference on Computer Vision, 2012.

35. Lin C, Chen WQ, Qiu C, Wu YF, Krishnan S, Zou Q. LibD3C: Ensemble Classifiers with a Clustering and Dynamic Selection Strategy. Neurocomputing, vol. 123, pp.424–435, 2014.

36. Avidan S. Support Vector Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, pp. 1064–1072, 2004. PMID: 15641735

37. Hare S, Saffari A, Torr PH. Struck: Structured Output Tracking with Kernels. IEEE International Conference on Computer Vision, 2011.

38. Ning JF, Yang JM, Jiang SJ, Zhang L, Yang MH. Object Tracking via Dual Linear Structured SVM and Explicit Feature Map. IEEE International Conference on Computer Vision and Pattern Recognition, 2016.

39. Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409–1422, 2012. doi: 10.1109/TPAMI.2011.239 PMID: 22156098

40. Saffari A, Leistner C, Santner J, Godec M, Bischof H. On-line Random Forests. IEEE International Conference on Computer Vision, 2009.

41. Babenko B, Yang M, Belongie S. Robust Object Tracking with Online Multiple Instance Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pp. 1619–1632, 2011. doi: 10.1109/TPAMI.2010.226 PMID: 21173445

42. Zou Q, Zeng JC, Cao LJ, Ji RR. A Novel Features Ranking Metric with Application to Scalable Visual and Bioinformatics Data Classification. Neurocomputing, vol. 173, pp.346–354, 2016.

43. Jiang N, Liu WY, Wu Y. Learning Adaptive Metric for Robust Visual Tracking.IEEE Transactions on Image Processing, vol. 20, no.8, pp.2288–2300, 2011. doi: 10.1109/TIP.2011.2114895 PMID: 21335312

44. Mei X, Ling H. Robust Visual Tracking using L1 Minimization. IEEE International Conference on Computer Vision, 2009.

45. Zhang T, Ghanem B, Liu S, Ahuja N. Low-rank Sparse Learning for Robust Visual Tracking. European Conference on Compute Vision, 2012.
46. Zitnick CL, Dollár P. Edge boxes: Locating Object Proposals from Edges. European Conference on Computer Vision, 2014.
47. Uijlings JRR, Sande KEA, Gevers T, Smeulders AWM. Selective Search for Object Recognition. International Journal of Computer Vision, vol. 104, no. 2, pp. 154–171, 2013.
48. Wang X, Yang M, Zhu S, Lin Y. Regionlets for Generic Object detection. IEEE International Conference on Computer Vision, 2013.
49. Girshick RB, Donahue J, Darrell T, Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. IEEE International Conference on Computer Vision and Pattern Recognition, 2014.
50. Hua Y, Alahari K, Schmid C. Online Object Tracking With Proposal Selection. IEEE International Conference on Computer Vision, 2015.
51. Liang PP, Liao CY, Mei X, Ling HB. Adaptive Objectness for Object Tracking. arXiv:1501.00909, 2015.
52. Zhu G, Porikli F, Li HD. Tracking Randomly Moving Objects on Edge Box Proposals. arXiv:1507.08085, 2015.
53. Yilmaz A, Javed O, Shah M. Object Tracking: A Survey. ACM Computing Surveys, vol. 38, no.4, pp.1–45, 2006.
54. Li X, Hu W, Shen C, Zhang Z, Dick A, Hengel A. A Survey of Appearance Models in Visual Object Tracking. ACM Transactions on Intelligent Systems and Technology, vol. 4, no.4, pp.1–58, 2013.
55. Sohn K, Zhou GY, Lee CS, Lee HL. Learning and Selecting Features Jointly with Point-wise Gated Boltzmann Machines. International Conference on Machine Learning, 2013.
56. Torralba A, Fergus R, Freeman W. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 11, pp. 1958–1970, 2008. doi: 10.1109/TPAMI.2008.128 PMID: 18787244
57. Isard M, Blake A. CONDENSATION-Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision, vol. 29, no.1, pp.5–28, 1998.
58. Lee HL, Grosse R, Ranganath R, Ng AY. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. International Conference on Machine Learning, 2009.
59. Dollár P, Zitnick CL. Structured Forests for Fast Edge Detection. IEEE International Conference on Computer Vision, 2013.
60. Gao J, Ling H, Hu W, Xing J. Transfer Learning based Visual Tracking with Gaussian Processes Regression. European Conference on Computer Vision, 2014.