ONLINE VIDEO TRACKING USING COLLABORATIVE CONVOLUTIONAL NETWORKS

Hao Guan, Xiangyang Xue, An Zhiyong

Shanghai Key Laboratory of Intelligent Information Processing,
School of Computer Science,
Fudan University, Shanghai, China
guanh13@fudan.edu.cn

ABSTRACT

Recently, convolutional neural network (CNN) models have achieved great success in many vision tasks. However, few attempts have been made to explore CNN for online model-free object tracking without time-consuming offline training. In this paper, we propose an online convolutional network (OCN) for visual object tracking. To make the network less dependent on labeled data, K-means is employed to learn multistage filter banks for hierarchical feature learning. To preserve more spatial information for tracking, down-sampling and pooling operations are eliminated, which enables our system more sensitive to spatial variations. A regression model is adopted as the output layer of OCN to predict the position changes of the target. To deal with the stability-plasticity dilemma, two OCNs with different update rates are integrated to construct an ensemble framework. Experiments on challenging benchmark video sequences demonstrate that the proposed tracker outperforms several state-of-the-art methods.

***Index Terms*—** Visual tracking, online learning, convolutional network, ensemble tracking

1. INTRODUCTION

Visual object tracking is a fundamental topic in computer vision with numerous applications including surveillance, robotics and human-computer interaction . Given the bounding box of an object in a certain video frame, the goal is to compute and output the bounding box of the target over the subsequent frames. Although many tracking methods have been proposed in the past decades [1, 2, 3], the task is still very challenging due to many factors such as illumination variation, occlusion as well as background clutter.

Most of the state-of-the-art tracking methods focus on designing new appearance models [4] which can effectively distinguish the target from its background. In [5], the authors used Haar-like features and multi-instance learning which could tolerate some false samples to track the target. In [6], the authors adopted local binary pattern (LBP) features and an online random fern classifier to detect the target. In [7], HOG features were introduced to distinguish the target from its background during tracking. In [8], a local sparse representation method was proposed within the particle filter framework. In [9], a sparse representation via the compressive sensing theory was proposed as the appearance model. Despite the diversity, most trackers rely on hand-crafted features which may not be good enough to handle more challenging video sequences.

In recent years, convolutional neural network (CNN) models [10] which can learn hierarchical features automatically from raw images have shown excellent performance on several tough vision tasks, such as image classification [11], object detection [12] and action recognition [13]. Despite such popularity, there are few attempts to apply CNN to online model-free object tracking. In [14], the authors employed a two-layer CNN which was pre-trained on auxiliary video sequences within a particle filter framework for tracking. In [15], the authors adopted a pre-trained deep CNN with a SVM classifier to learn a saliency map which is helpful for locating the target. In [16], two CNNs trained by 20000 image pairs were used for human tracking.

However, there are mainly three drawbacks of these methods. First, using pre-trained models for online model-free tracking may not be proper and the effectiveness of appearance model is greatly influenced by the auxiliary dataset for offline training. Second, the down-sampling operations in conventional CNN may lose some spatial information which is essential for tracking. Third, most of the CNN models for tracking adopt binary classifiers to separate the target from background which is less robust due to false samples.

In this paper, we focus on designing a new CNN model which is more suitable for online tracking and avoids the need for massive training datasets and time. Specifically, we employ K-means to learn the data-adapting convolutional filter bank followed by nonlinearity in each layer of a typical CNN, which makes the network easy to train with much less training time and labeled data. To preserve more spatial information, down-sampling and pooling operations are eliminated,

This work was supported in part by NSFC under Grant 61572138, two STCSM's programs (No. 15511104402 and No. 15JC1400103), and EU FP7 QUICK Project under Grant Agreement (No. PIRSESGA2013-612652).

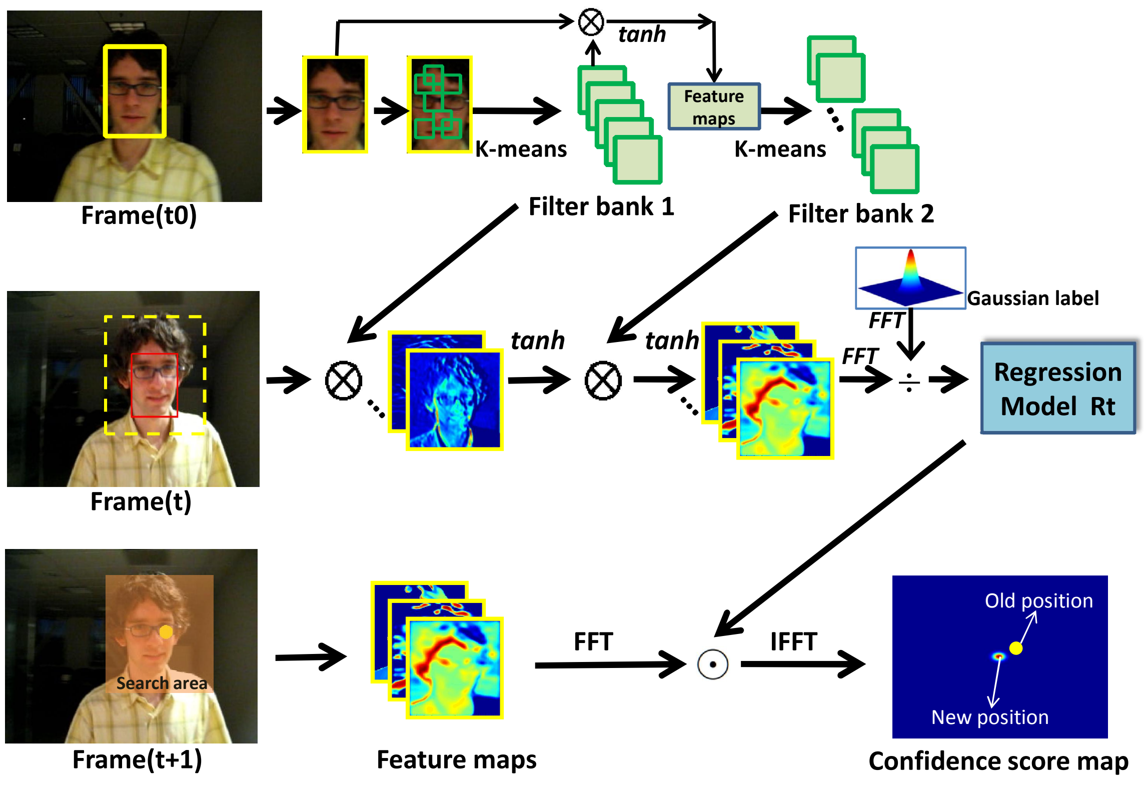

Fig. 1. Overview of the proposed tracking algorithm. At frame t, the feature maps of the target are learned by the multistage hidden layers of the network which are trained in unsupervised manners (K-means). The regression layer is trained by the feature maps and used in frame $t + 1$ to compute the score map whose peak point indicates the new position of the target.

which makes our network more sensitive to position changes. In addition, we adopt a regression model rather than a binary classifier as the output layer of the OCN network which can output a pixel-wise response score map whose peak point indicates the position of the target. We name the new CNN model for tracking as OCN and the tracker as OCNT. To tackle the stability-plasticity dilemma, we construct an ensemble framework using two OCNs with differentially-paced update. One OCN updates aggressively to adapt for large appearance changes while the other one updates conservatively to keep a stable appearance model. Through the collaboration of the two OCNs, the OCNT tracker can achieve robust performances.

Our contributions mainly lie in: (1) we propose a novel CNN model (OCN) which is deliberately designed for online video tracking. The OCN can learn hierarchical unsupervised features of the target without massive training datasets or time. (2) We adopt a regression method rather than a binary classifier to output a pixel-wise response score map which greatly alleviates the drift problem caused by false samples. (3) We construct an ensemble framework using two collaborative OCNs which makes the tracker both adaptive to appearance changes and robust to potential error accumulation.

2. VIDEO TRACKING VIA ONLINE CONVOLUTIONAL NETWORK

2.1. System overview

The OCN can be divided into two parts, the feature-learning part and the location regression part. In the feature-learning stage, we adopt K-means to learn the filter bank of each layer which can get hierarchical object features. The $tanh$ function is used as the nonlinearity. Note that down-sampling operations are eliminated between consecutive layers. The output layer of OCN is based on a regression model which is trained by the learned feature maps and can output a pixel-wise response score map of the whole search area whose peak indicates the position of the target during online tracking. Figure 1 shows the main flow of our tracking system.

2.2. Hierarchical unsupervised feature learning of OCN

A typical CNN involves multiple learning layers stacked on the top of each other which are trained by stochastic gradient descent algorithm. Learning a network of this type in supervised manners needs a lot of labeled training data and time which cannot be satisfied in online tracking. To deal with this, the OCN network is deliberately designed for online tracking. Specifically, the multistage hidden layers of OCN are trained by unsupervised learning methods. In practice, K-means is employed to learn the data-adapting convolution filter banks. K-means has been proven to be a very effective feature learning algorithm [17]. The advantage of K-means is that it does not require labeled data for training and the only parameter is the number of centroids.

Given the bounding box of the target in a certain frame at time t_0, the image patch within the bounding box can be denoted as I_{t_0}. We can collect a total of n sub patches of I_{t_0} as training samples in a sliding window manner. Note that each sub patch has the same size with the convolution filter to be learned. Then each sub patch is preprocessed by subtracting mean value and dividing by the standard deviation of its elements which correspond to local brightness and contrast normalization. So we can get n normalized patches:

$$P = [\bar{p}_1, \bar{p}_2, \bar{p}_3, ..., \bar{p}_n] \tag{1}$$

where $\bar{p}_i$ is a normalized patch. After preprocessing, K-means cluttering is applied to the normalized patches. Let N_i denote the number of filters in layer i. The filter bank V in the first layer can be learned by K-means clustering on the normalized patches:

$$V = \{v_i\}, i = 1, 2, 3, ..., N_1, \tag{2}$$

After the filters have been learned in the first layer, we can get the outputs of the filters (feature maps):

$$X_i = I_{t_0} \otimes v_i, i = 1, 2, 3, ..., N_1, \tag{3}$$

where $\otimes$ represents 2D convolution operator. After the convolution computation, there follows a nonlinear operation which is the final step of each layer. We use $tanh$ function as the nonlinearity.

In the second layer, K-means clustering is applied to all the sub patches of the N_1 feature maps and get N_2 convolution filters.

The process above can be repeated and a deeper architecture can be built if necessary. All the learned convolution filters in each stage can be used to extract hierarchical features of the target automatically.

2.3. Regression layer of OCN for online tracking

After the convolution filters of OCN have been learned, the next step is to train a classifier which is adopted to locate the target during tracking. Many discriminative trackers train a binary classifier to separate the target from background where **1** indicates target and **0** for background. Positive samples are defined as the patches that have high overlap rate (larger than a threshold) with the target while negative samples are the ones that have low overlap rate (smaller than a threshold). Since it is not clear how the thresholds should be estimated in an online learning framework, it is very easy to bring false samples which may fool the classifier and lead to drift. A better approach is to use a regression model which assign weak labels to samples. A weak label is defined as a continuous real value between 0 and 1 according to the distance between the sample and the target center. This strategy can prevent from manually defining thresholds for 0/1 labels, thus can resist to some potential errors brought by false samples.

Specifically, let R denote a search area with the size $M \times N$ which is centered by the target bounding box. The patch in the target bounding box is denoted as $\mathbf{x}$. Sampling is performed in R in a sliding window manner, then all the training samples are generated as the cyclic shift versions of $\mathbf{x}$, denoted as $\mathbf{x}_i$ where $i \in \{0, 1, ..., M-1\} \times \{0, 1, ..., N-1\}$. Note that each training sample is assigned a weak labels $y_i \in [0, 1]$ in terms of the shifted distance to the target center. A regression function $f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle$ is trained using the samples by minimizing the regression error:

$$\min_{\mathbf{w}} \sum_i \| \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - y_i \|^2 + \lambda \|\mathbf{w}\|^2 \tag{4}$$

where $\phi(\star)$ is a nonlinear function and $\lambda \geq 0$ is a regulation parameter. Defining a RBF kernel function $g(x, x') = \langle \phi(x), \phi(x') \rangle$, the regression model can be derived as $f(\mathbf{x}) = \sum_i \alpha_i g(\mathbf{x}_i, \mathbf{x})$ where α is the dual variable of $\mathbf{w}$ and it has a closed form solution:

$$\boldsymbol{\alpha} = (G + \lambda I)^{-1} \mathbf{y} \tag{5}$$

where G is a kernel matrix. (5) is a general solution for the regression layer of our OCN. The RBF kernels are updated updated online to adapt for appearance changes of the target. In other words, the RBF kernels can have some memory of the target.

Due to the circulant property of the training samples, Fast Fourier Transform can be introduced to speed up the computation . Let the Discrete Fourier Transform be denoted as a vector with a hat, e.g., $\hat{\alpha} = \mathscr{F}(\alpha)$. According to [18], if the kernel function $g(\star)$ is shift invariant, e.g., a RBF kernel, $\hat{\alpha}$ can be derived based on the properties of circulant matrices[19]:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{g}}^{\mathbf{xx}} + \lambda} \tag{6}$$

where $\mathbf{g}^{\mathbf{xx}}$ is a vector whose ith element is $g(\mathbf{x}_i, \mathbf{x})$.

When a new video frame arrives, a patch $\mathbf{z}$ with the same size of $\mathbf{x}$ is cropped out. Note that $\mathbf{z}$ is centered by using the tracking result (bounding box) of the last frame. Then the response score map for the whole search area is computed as:

$$f(\mathbf{z}) = \mathscr{F}^{-1}(\hat{\mathbf{g}}^{\mathbf{xz}} \odot \hat{\boldsymbol{\alpha}}) \tag{7}$$

where $\odot$ denotes the element-wise product. $f(\mathbf{z})$ is the response for all the cyclic versions of $\mathbf{z}$ and the new position of the target is determined by searching for the location of the maximal value of $f(\mathbf{z})$.

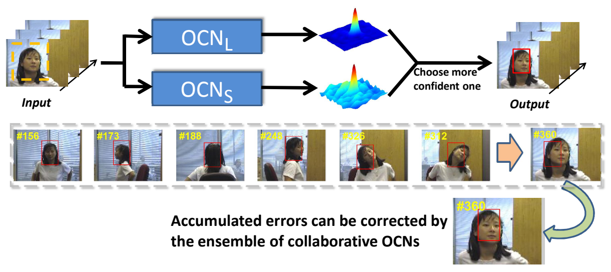
Input
OCN_L
OCN_S
Choose more confident one
Output
#156
#173
#188
#248
#262
#282
#360
Accumulated errors can be corrected by the ensemble of collaborative OCNs
#360

Fig. 2. The ensemble of collaborative OCNs. Though OCN$_S$ with aggressive update can handle drastic appearance changes, it may accumulate errors. As a compensation, OCN$_L$ is more stable to resist these potential errors.

3. ENSEMBLE FRAMEWORK WITH LONG AND SHORT TERM MEMORIES

Online visual tracking faces a stability-plasticity dilemma. If the tracker does not update enough, it may not adapt well to appearance changes. However, if the tracker just holds short-term memory and update very frequently, it may accumulate errors and suffer some drift.

To deal with this dilemma, we construct an ensemble framework. After the feature maps have been learned, the final output is determined by the RBF kernel functions. The RBF kernel are updated online to adapt for appearance changes, which indicates that the whole neural network can hold some memory. Motivated by this, we design a framework involving two OCNs with long and short term visual memories. The basic idea is to make one OCN (OCN$_S$) account for short-term memory while the other OCN (OCN$_L$) hold long-term memory. During tracking, OCN$_S$ updates aggressively to adapt for dramatic appearance changes while (OCN$_L$) updates conservatively to be resistant to potential error accumulation. The final estimation is determined by the more confident one. Through the collaboration of the two OCNs, the whole tracking system can have a more robust performance.

We first present the update strategy of OCN$_S$. The OCN$_S$ updates at every time step. Due to the aggressive update, OCN$_S$ can only hold memory of the appearance in a short-term span. Let $\hat{\mathbf{x}}$ denote the RBF kernel center (the stored appearance model). The update is computed with a rate η as:

$$\hat{\mathbf{x}}_t = (1-\eta)\hat{\mathbf{x}}_{t-1} + \eta \mathbf{x}^t \tag{8}$$

where t denotes the index of the current frame. The update rate η in (8) is set to a relatively large value which makes OCN$_S$ aggressive for drastic appearance changes. As a compensation, the OCN$_L$ mainly accounts for stable appearances. It adopts a conservative update strategy as:

$$\hat{\mathbf{x}}_n = (1-f)\hat{\mathbf{x}}_{n-1} + f\mathbf{x}_r \tag{9}$$

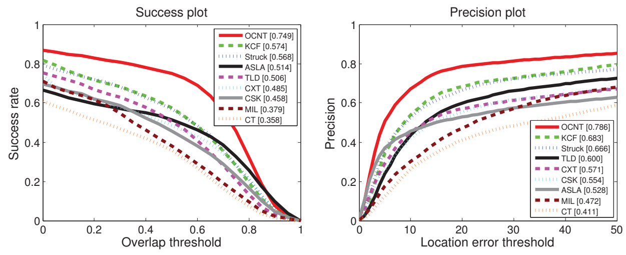

Fig. 3. Overall performance of the proposed OCNT tracker and several state-of-the-art trackers on 50 challenging video sequences from the CVPR13 benchmark.

where f is a fixed small value for conservative update. Note that $\mathbf{x}_n$ is used here instead of $\mathbf{x}_t$, this is because the OCN$_L$ does not update at every frame but when there is a highly confident patch $\mathbf{x}_r$ whose confidence score is larger than a threshold.

We adopt a measurement of peak strength called Peak to Sidelobe Ratio (PSR) to define the confidence score. A higher PSR value indicates a less ambiguity about the decision, thus the tracker is more confident about the estimation. Mathematically, the confidence score $conf$ based on PSR is defined as:

$$conf = \frac{\max f(\mathbf{z}) - \mu}{\sigma} \tag{10}$$

where μ and σ are the mean and the standard deviation of the response score map $f(\mathbf{z})$ of OCN respectively.

4. EXPERIMENTS

To evaluate the performance of the proposed OCNT tracer, we test it on a recent benchmark [2] which is widely used for tracking. The OCNT tracker is tested on 50 challenging video sequences that involve different tough attributes for tracking, such as illumination variation, occlusion, fast motion, etc. Some state-of-the-art trackers are used for comparison. These trackers include KCF tracker[7], Struck tracker [20], TLD tracker[6], CXT tracker[21], ASLA tracker[8], C-SK tracker[18], CT tracker[9] and MIL tracker [5].

4.1. Implementation details

Our OCNT tracker is implemented in MATLAB on a regular PC with an Intel Core i7 CPU and 8 GB RAM. For all reported experiments, the number of layers of OCN is set to 2, the size of filters of each stage is set to 5×5 and the number of filters of each stage is set to 8. As for the K-means clustering, all the filters are randomly initialized following standard Gaussian distribution and the epoch of clustering is set to 50. The update rate of OCN$_S$ is set to 0.075 and the update rate of OCN$_L$ is set to 0.001.

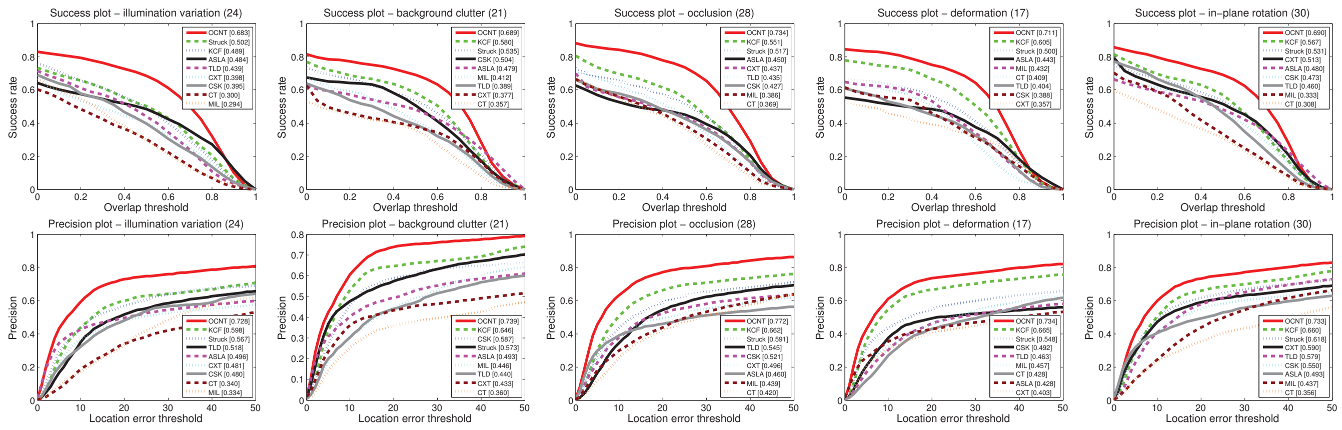

Fig. 4. Success plots and precision plots of five different attributes including: illumination variation, background clutter, occlusion, deformation and in plane rotation.

4.2. Evaluation methodology

We follow the evaluation protocols in the benchmark[2] where the success plot and precision plot are adopted to e-valuate the overall performance of trackers. The success plot indicates the ratios of successful frames whose overlap rate is larger than a given threshold. By varying the threshold gradually from 0 to 1, it gives a plot of the success rate against the overlap threshold for each tracker. The precision plot indicates the percentage of frames whose center location error is within a given threshold. The center location error is the Euclidean distance (in pixels) between the center of the tracking result and the ground truth for each frame. We rank the tracking algorithms based on Area Under Curve (AUC) for success plot and center location error at 20 pixels for precision plot.

4.3. Overall evaluation

The overall comparison results of OCNT and the other trackers on benchmark are shown in Figure 3. The performance score of each tracker is shown in the legend of each plot.

It is observed from Figure 3 that our OCNT tracker, KCF tracker and Struck tracker achieve good performances. Overall, our OCNT tracker outperforms the state-of-the-art trackers in terms of both overlap rate and location precision.

The robustness of our OCNT tracker can be attributed as follows. First, the hierarchical features learned by K-means are more discriminative than traditional hand-crafted features such as Haar features or binary pattern features. Second, We use regression rather than binary classifiers to estimate the location, which has effectively prevent the false samples from contaminating the online appearance model. Third, the ensemble framework further improves the robustness of the tracker by making a balance between stability and plasticity.

4.4. Attribute based evaluation

A tracker performs well on one video sequence may show bad performance on another due to different challenging factors. In the tracking benchmark [2], all the 50 sequences are annotated with 11 different attributes to describe the typical challenges for tracking, e.g., illumination variation, occlusion, background clutters, etc. These attributes are useful for analyzing and evaluating the performance of a tracker in d-ifferent scenarios. We report results with success plots and precision plots for five typical challenging attributes in Figure 4.

On the illumination variation subset, our OCNT tracker outperforms the second best trackers by a large margin which indicates that the learned hierarchical features by OCN is less vulnerable to illumination changes. On the background clutter subset, both our OCNT tracker and KCF tracker outperfor-m others, which can be attributed to their better discriminative power. On the occlusion subset, OCNT tracker achieves the best performance, this may benefit from the ensemble framework which can resist to potential errors from the occlusion. On the deformation and in-plane rotation subsets, both our OCNT tracker and KCF tracker perform better than other trackers which indicates that the learned features with regression method are more robust to dramatic appearance changes such as deformation and rotation.

5. CONCLUSION

In this paper, we present a novel online convolutional network (OCN) for tracking. The proposed OCN learns hierarchical features of the target by cascade filters learned by K-means and down-sampling operations are eliminated to preserve more spatial information. In addition, a regression layer instead of a binary classifier is adopted which shows robustness. We demonstrate that the hierarchical unsupervised features learned by OCN are capable of handling different challenging factors for tracking and the biggest advantage is that it avoids huge labeled training data and time. Furthermore, an ensemble framework using two OCNs with differenly-paced update makes the whole tracking system both adaptive to drastic appearance changes and robust to drift. Experiments on the tracking benchmark demonstrate the effectiveness and robustness of the proposed tracker.

6. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[2] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[3] W.M.S. Smeulders, D. M. Chu, R. Cucciara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: A experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[4] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 58, 2013.

[5] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[6] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[7] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[8] J. Xu, H. Lu, and M. H. Yang., "Visual tracking via adaptive structural local sparse appearance model," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[9] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proc. of European Conference on Computer Vision*, 2012.

[10] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *IEEE International Symposium on Circuits and Systems*, 2010.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[13] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014.

[14] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 11424–1435, 2015.

[15] H. Seunghoon, Y. Tackgeun, K. Suha, and H. Bohyung, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. of The 32nd International Conference on Machine Learning*, 2015.

[16] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010.

[17] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," *AISTATS*, vol. 15, no. 14, pp. 215–223, 2011.

[18] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploring the circulant structure of tracking-by-detection with kernels," in *Proc. of European Conference on Computer Vision*, 2012.

[19] R. M. Gray, *Toeplitz and circulant matrices: A review*, Now Publishers Inc., 2006.

[20] S. Hare, A. Saffari, and P.H.S. Torr, "Structured output tracking with kernels," in *Proc. IEEE International Conference on Computer Vision*, 2011.

[21] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2011.