



# Hierarchical Convolution Feature for Target Tracking with Kernel-Correlation Filtering

Jing Zhang<sup>3</sup>(✉), Dong Hu<sup>1,2,3</sup>, Biqu Zhang<sup>3</sup>, and Yuwei Pang<sup>3</sup>

<sup>1</sup> Education Ministry's Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing, China

<sup>2</sup> Education Ministry's Engineering Research Center of Ubiquitous Network and Health Service, Nanjing, China

<sup>3</sup> Jiangsu Province's Key Lab of Image Processing and Image Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China  
{1217012312, hud}@njupt.edu.cn

**Abstract.** Target tracking is widely used in many fields, but tracking performance still needs to be improved due to factors such as deformation, illumination and occlusion. In this paper, we propose a scale adaptive target tracking solution based on hierarchical convolution features and establish a kernel correlation filtering target tracking framework that combines multi-layer convolution features. The improved convolutional neural network is used to extract multi-layer features, and the correlation filters of each layer are separately trained to perform weighted fusion to obtain the target position. Then, the edge box algorithm is adopted to obtain the size of the actual tracking frame to achieve exact target tracking. An extensive evaluation on OTB-2013 with public test sequences are conducted. Experimental results and analysis indicate that our method is better than other known advanced tracking algorithms even in video sequences with many uncertain factors, while the speed and accuracy of tracking can be effectively improved.

**Keywords:** Object tracking · Convolution neural network · Kernel correlation filter · Edge boxes

## 1 Introduction

Visual tracking is a fundamental computer vision task with a wide range of applications. Although much progress has been made in the past decade, tremendous challenges still exist in designing a robust tracker that can well handle significant appearance changes, pose variations, severe occlusions and background clutters. In order to adapt to the actual application scenarios and ensure the accuracy and robustness of target tracking, lots of efforts have been paid in theoretic studies and applications.

Existing appearance-based tracking methods adopt either generative or discriminative models. The generative methods establish a model for the target region in the current frame, and then match the next frame, such as particle filter [1], Kalman filter [2] and so on. Discriminative methods treat the tracking problem as a two-category

problem that find the decision boundary of the target and background. For example, the Stuck algorithm [3] and the Compressed Tracking (CT) algorithm [4]. Whereas, this kind of tracking algorithm is slow. Subsequently, correlation filtering algorithms are introduced into the target tracking. For example, the Circulant Structure of Tracking-by-detection with Kernels (CSK) algorithm [5] opens the beginning of correlation filtering research; Kernel Correlation Filter (KCF) [6] uses the Histograms of Oriented Gradients (HOG) feature to convert a single channel into multiple channels. Although this algorithm is faster, the disadvantage is that the target rotation and occlusion problems cannot be solved.

Driven by the emergence of large-scale visual data sets and fast development of computation power, Deep Neural Networks (DNNs), especially Convolutional Neural Networks (CNNs), with their strong capabilities of learning feature representations, such as Hierarchical Convolutional Features (HCF) [7]; Recursive neural networks are used to track targets, such as Recurrently Target-Attending Tracking (RTT) [8]. But how to design neural networks and tracking processes to achieve speed improvement, there is still a lot of research space.

Considering the robustness and real-time of tracking, in our method, the multi-layer depth feature is extracted by convolutional neural network. Based on the hierarchical convolution feature and KCF target tracking, the tracking problem is deeply studied in complex environment.

The rest of the paper is organized as follows. The Sect. 2 is the related technical analysis of convolutional neural network and kernel correlation filters. The Sect. 3 discusses the detailed algorithm, the scale adaptive target tracking method based on layered convolution characteristics. The Sect. 4 provides a discussion concerning experimental simulation. Finally, the summary is delivered in Sect. 5.

## 2 Related Work

In our algorithm, the features of the convolutional neural network are applied to the kernel correlation filter tracking framework. Therefore, this section first introduces the principle of convolutional neural networks, and secondly introduces the kernel correlation filter tracking algorithm.

### 2.1 Convolutional Neural Network

Convolutional neural network is a multi-layer neural network. The whole structure includes convolution layer, nonlinear activation function, pooling layer. The high-level information is obtained from the original data layer by layer. The main role of the convolutional layer is to use the convolution kernel for feature extraction and feature mapping. The input image is first convoluted with the convolution kernel, and the result is used as the input to the nonlinear activation function. The activation function is used to add nonlinear factors. The commonly used activation functions are Tanh function, ReLU function and so on. The pooling layer has the effect of quadratic feature

extraction, which can reduce the dimension of the feature map. The typical pooling operation is the average pooling and the largest pooling. The fully connected layer is a classifier of the convolutional neural network. For classification tasks, SVM is usually used because it can be combined with CNN to solve different classification tasks.

Among commonly used convolutional neural networks, e.g. AlexNet, VggNet, and ResNet, we use VGGNet-19 [9] network, because it is easy to migrate to other image recognition projects. Besides, VGGNet trained parameters can be download for a good initialization weight operation.

### 2.2 Tracking by Kernel Correlation Filters

The kernel correlation filtering algorithm is to train better classifiers to find the decision boundary of the target and background. The purpose of training is to find a function  $f(x) = w^T x$  that minimizes the error function. The objective function can be expressed as Eq. (1):

$$\min_w \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2 \tag{1}$$

where  $\lambda$  is a regularization coefficient used to control overfitting. We can get the solution in complex domain  $w = (X^H X + \lambda I)^{-1} X^H y$ , where the matrix  $X$  has one sample  $x_i$  per line, each element of  $y$  is a regression target  $y_i$ ,  $X^H$  represents a complex conjugate transpose matrix, and  $I$  represents the identity matrix. Using the properties of the diagonalization of the circulant matrix to obtain the simplified ridge regression of the Fourier diagonalization, the following formula  $\hat{w} = \text{diag}(\frac{\hat{x}^*}{\hat{x}^* \otimes \hat{x} + \lambda}) \hat{y}$  is obtained, where  $\hat{x}^*$  is the complex conjugate of  $\hat{x}$ ,  $\otimes$  represents the dot multiplication of the element. Since most of the cases are nonlinear, high-dimensional solutions and kernel functions [10] have been introduced. The objective function can be expressed as:

$$f(z) = w^T z = \sum_{i=1}^n \alpha_i \kappa(z, x_i) \tag{2}$$

After the ridge regression is nucleated, the form of the frequency domain solution is  $\hat{\alpha} = \frac{\hat{y}}{k^{xx} + \lambda}$ , where  $k^{xx}$  represents the kernel correlation of any two vectors  $x$  and  $x'$ , the symbol  $\hat{\cdot}$  represents the DFT transform of the vector. It is easy to prove that the kernel matrix between all training samples and all candidate image blocks satisfies the condition of the cyclic matrix, so that the regression function in the frequency domain of all candidate image blocks can be obtained:  $\hat{f}(z) = \hat{k}^{xz} \otimes \hat{\alpha}$ . In particular, when the kernel function is a Gaussian kernel, we can get Gaussian kernel related  $k^{xx'} = \exp(-\frac{1}{\sigma^2} (||x||^2 + ||x'||^2 - 2F^{-1}(\hat{x}^* \otimes \hat{x}')))$ . By the maximum value of the positioning, the relative motion of the tracking target can be obtained.

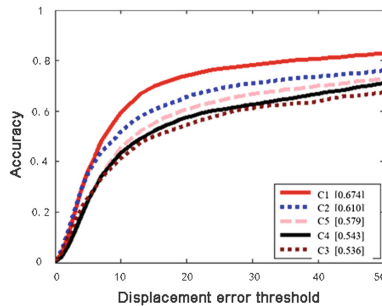
### 3 Scale Adaptive Tracking Based on Hierarchical Convolution Features

In this section, we will give a detailed description of the algorithm we proposed. This section will be divided into two modules. The first module is used for target positioning. We extract improved multi-layer convolution features for VGG-Net and combine them with kernel correlation filtering algorithms for target localization. The second module is used for target scale estimation. We use the edge box algorithm to obtain the size of the actual tracking frame and achieve scale adaptation.

#### 3.1 Target Position Estimation

Considering the robustness and accuracy, it is proposed to combine the multi-layer convolution feature with the kernel correlation filtering algorithm to achieve the target position estimation.

The network model used here is VGGNet-19, including 5 sets of convolutional layers, a total of 16 layers, and the last three are fully connected layers. In order to deeply understand the characterization ability of each layer feature of the convolutional neural network, the single-layer convolution feature is applied to the KCF correlation filtering tracking algorithm respectively, and the tracking results are compared and analyzed. The experiment performed on 35 color video sequences in the OTB-2013. The result is expressed in terms of precision, which is calculated as the center point of the target position estimated by the tracking algorithm and the center point of the manually labeled target. For a given threshold, the distance between the two is less than the percentage of the video frame, and the general threshold is set to 20 Pixel.



**Fig. 1.** OPE precision map using different convolution layers

Figure 1 is an OPE accuracy map obtained using different convolution features. It can be seen that the first layer convolution feature has high resolution and can accurately locate the target, the fourth layer and the fifth layer convolution feature contain more semantic information and can roughly locate the target. Therefore, we use Conv1-2, Conv4-4, Conv5-4 layer convolution features for correlation filtering target tracking algorithm. We take the output of each convolutional layer as a multi-channel feature

[5], and take the virtual samples obtained from all the cyclic shift of feature X as training samples. Each cyclic shift sample has a corresponding Gaussian distribution label  $y_{ij} = e^{-\frac{(i-M/2)^2 + (j-N/2)^2}{2\sigma^2}}$ , where  $\sigma$  is the kernel width. Learn about correlation filters of the same  $x$  size by addressing the following minimization issues:

$$W^* = \arg \min_w \sum_{m,n} \|W \cdot X_{m,n} - y(m,n)\|^2 + \lambda \|W\|_2^2 \quad (3)$$

where linear product is defined as  $w \cdot x_{ij} = \sum_{d=1}^D w_{ijd}^D x_{ijd}$ . The filter learned in the frequency domain on the  $d$ th ( $d \in \{1, \dots, D\}$ ) channel is:

$$W^d = \frac{Y \odot \bar{X}^d}{\sum_{n=1}^D X^i \odot \bar{X}^i + \lambda} \quad (4)$$

where Y is the Fourier transform form of  $y_{ij}$ , the horizontal line on the letter indicates the complex conjugate. Let  $z$  be expressed as the feature vector on the  $l$ th layer and the size is  $M \times N \times D$ , and then the  $l$ th correlation response map can be calculated by the following formula:

$$f(z) = F^{-1} \left( \sum_{d=1}^D W^d \odot Z^d \right) \quad (5)$$

where operator  $F^{-1}$  represents inverse FFT transform. Let  $(\hat{m}, \hat{n}) = \arg \max_{m,n} f_l(m,n)$  denote the position of the maximum value on the  $l$ th layer, then the best position of the target in the  $l$ -1th layer is expressed as:

$$\begin{aligned} & \arg \max_{m,n} f_{l-1}(m,n) + \gamma f_l(m,n) \\ & s.t. \quad |m - \hat{m}| + |n - \hat{n}| \leq r \end{aligned} \quad (6)$$

The constraint indicates that only the region centered at  $(\hat{m}, \hat{n})$  and  $r$  is the radius is searched for in the in the  $l$ -1th layer correlation response graph. The response value from the latter layer is weighted as a regularization term and then propagated back to the response graph of the previous layer. In this way, the maximum value in the response graph of the last layer is the predicted position of the target.

### 3.2 Target Scale Adaptation

Based on target position estimation, our method proposes a scale adaptive target tracking by edge frame detection algorithm [11]. The edge frame detection algorithm traverses the entire image in a sliding window manner, and scores the bounding box of each sample, selects the top 200 candidate frames with the highest score, and performs a convolution operation on the candidate frame and the filter to obtain a response

graph. The maximum response value in the candidate target can be expressed as  $f_{\max} = \max(f_{\max,1}, f_{\max,2}, \dots, f_{\max,n})$ , where  $f_{\max,1}, f_{\max,2}, \dots, f_{\max,n}$  is the maximum response value in the response graph of each candidate target,  $n$  is the number of candidate targets. If  $f_{\max}$  less than  $f_p$  ( $f_p$  is the maximum response of the correlation filter by using the layered convolution feature), this means that the detection algorithm is most likely to find that the position of the target is not as accurate as the target position estimated by the convolution feature. Thus, abandoning the candidate target detected by the detection algorithm, and the target size remains unchanged. Otherwise updating the position and size using the damping factor  $\gamma$ . The scale update method is as follows:

$$(w_t, h_t) = \begin{cases} (w_{t-1}, h_{t-1}) + \gamma[(w_{p,t}, h_{p,t}) - (w_{t-1}, h_{t-1})], & \text{if } f_{\max} > f_p \\ (w_{t-1}, h_{t-1}), & \text{if } f_{\max} < f_p \end{cases} \quad (7)$$

where  $w_{t-1}$ ,  $h_{t-1}$ ,  $w_{p,t}$ ,  $h_{p,t}$  respectively indicate the width and height of the  $t$ -1th candidate frame with the largest response value in the target and the  $t$ -th frame, which  $\gamma$  is set to 0.5 as the learning rate. The target location is updated as follows:

$$I_t = \begin{cases} I_{d,t} + \gamma(I_{p,t} - I_{d,t}), & \text{if } f_{\max} > f_p \\ I_{d,t}, & \text{if } f_{\max} < f_p \end{cases} \quad (8)$$

where  $I_{d,t}$  is the target position based on the hierarchical convolution feature,  $I_{p,t}$  is the target position of the maximum response value corresponding to the  $t$  frame. Finally, the location and size of the target are estimated to achieve target tracking.

## 4 Experiments

In this section we evaluate our algorithm from two aspects. Firstly, a qualitative comparison is provided, we display the tracking effect for scale change and occlusion test sequence on OTB-2013 dataset. Secondly, through the quantitative analysis, the tracking effects of several excellent open source trackers in the visual tracker benchmark test were compared.

The simulation environment for this experiment is MATLAB, and the experimental environment is on an i7 machine with 8 GB of memory. To assess accuracy and success rate, we compared four advanced tracking algorithms in literature: HCF, Stuck, KCF, CT. In these experiments, three video sequences in the standard target tracking library OTB-2013 were tested.

### 4.1 Qualitative Experiment Verification

In this section, we select three video sequences from OTB-2013 for qualitative analysis, which are Dog1, Singer1, and CarScale video sequences.

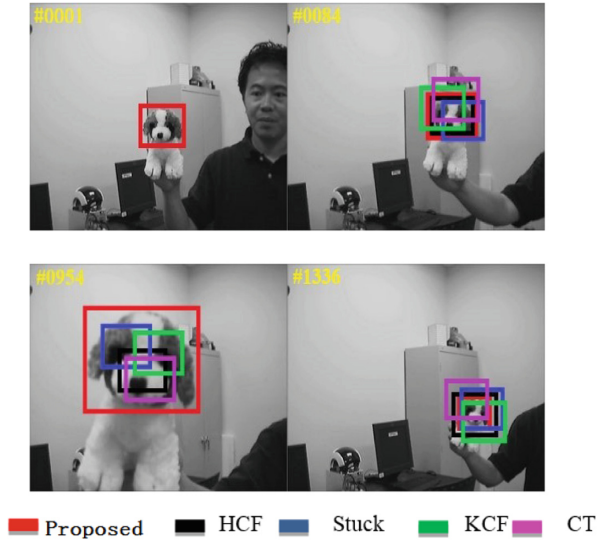


Fig. 2. Dog1 video sequence renderings

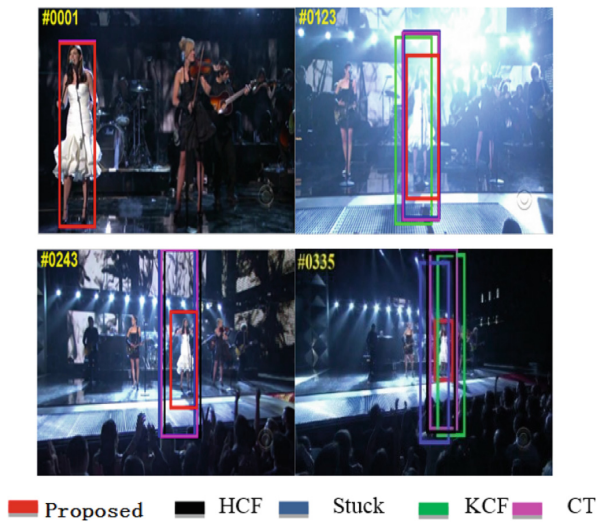
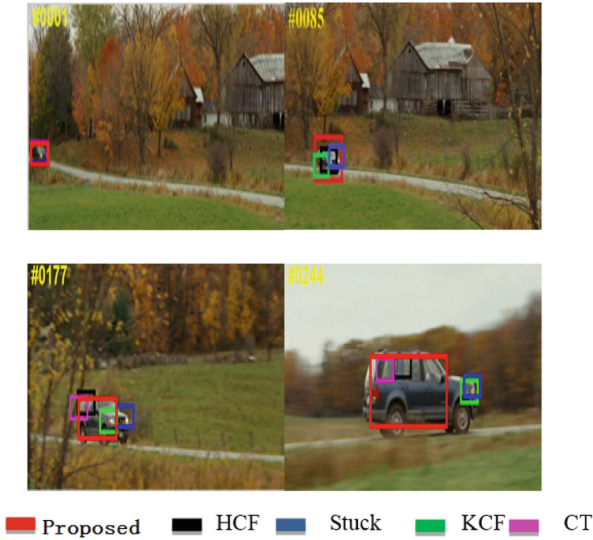


Fig. 3. Singer1 video sequence renderings



**Fig. 4.** CarScale video sequence renderings

As shown in Figs. 2, 3 and 4, there are obvious scale changes in the three video sequences. If the target becomes larger, the sample will lose some important information. HCF, CT, Stuck and KCF can only track a small part of the target. Our algorithm can track the target accurately and achieve the scale adaptation.

## 4.2 Quantitative Experiment Verification

In this section, we quantitatively analyze the algorithm in the Visual Tracker Benchmark and compare it with several popular algorithms, as shown in Fig. 5. It can be seen from (a) and (b) that compared with other algorithms, the two indicators of the algorithm have the best results, the average accuracy reaches 81.2%, which is 0.3% higher than HCF; the average success rate reached 65.8%, an increase of 9.8% compared to HCF. It can be seen from (c) and (d) that the proposed algorithm achieves better tracking results in 28 scale-changing video sequences compared to other algorithms. The average accuracy is improved by 2.6% and the average success rate is increased by 14.6%. It shows that the proposed algorithm has better robustness and can better adapt to changes in target scale.



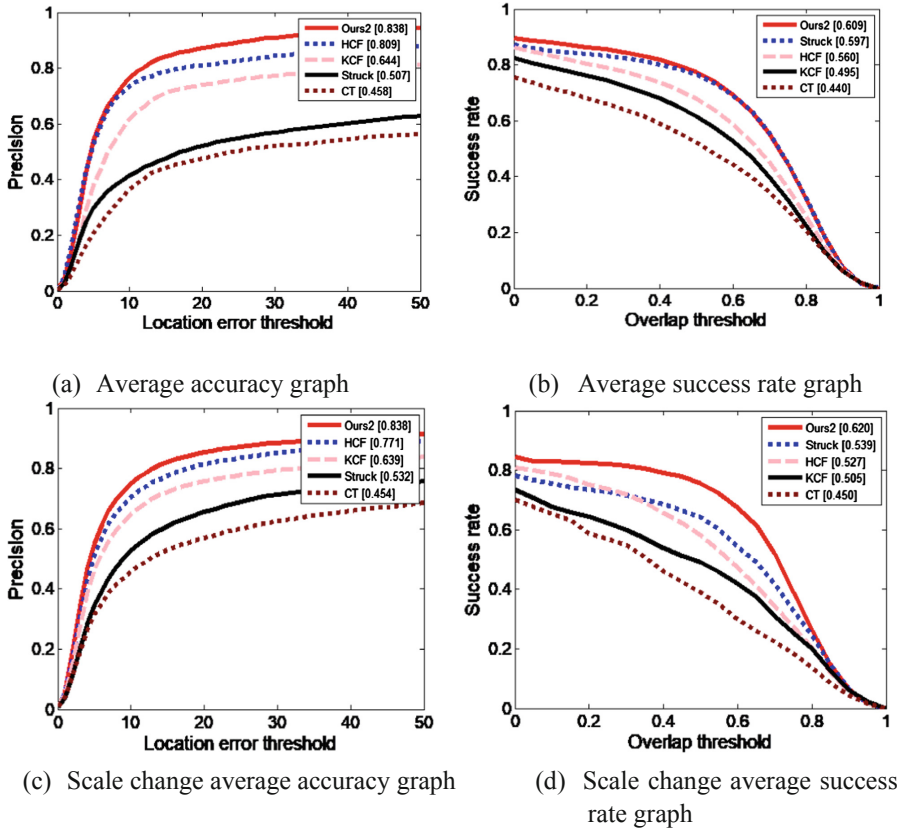


Fig. 5. Tracking performance comparison chart

## 5 Conclusions

In this paper, we empirically present some important properties of CNN features under the viewpoint of visual tracking. Based on these attributes, we propose a tracking algorithm for pre-training image classification tasks using complete convolution network. The improved convolutional neural network extraction feature is applied to the kernel correlation filtering tracking algorithm to achieve accurate target location. At the meanwhile, the edge frame detection algorithm is used to generate the target positional bounding box. The problem of fast scale change in target tracking is solved, and the accuracy and robustness of target tracking are improved.

## References

1. Chang, C., Ansari, R.: Kernel particle filter for visual tracking. *IEEE Signal Process. Lett.* **12**(3), 242–245 (2005)
2. Torkaman, B., Farrokhi, M.: Real-time visual tracking of a moving object using pan and tilt platform: A Kalman filter approach. In: 20th Iranian Conference on Electrical Engineering, Iran, pp. 56–67 (2012)
3. Hare, S., Saffari, A., Torr, P.H.S.: Struck: structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(10), 2096–2109 (2015)
4. Zhang, K., Zhang, L., Yang, M.-H.: Real-Time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012*. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-33712-3\\_62](https://doi.org/10.1007/978-3-642-33712-3_62)
5. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012*. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-33765-9\\_50](https://doi.org/10.1007/978-3-642-33765-9_50)
6. Henriques, J.F., Caseiro, R., Martins, P., et al.: High-Speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 583–596 (2015)
7. Ma, C., Huang, J.B., Yang, X., et al.: Hierarchical convolutional features for visual tracking. In: *IEEE International Conference on Computer Vision 2015*, pp. 111–121. IEEE, Chile (2015)
8. Cui, Z., Xiao, S., Feng, J., et al.: Recurrently target-attending tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition 2016*, pp. 1449–1458. IEEE, LAS VEGAS (2016)
9. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*, pp. 569–577 (2014)
10. Schölkopf, B., Smola, A.: Learning with kernels: support vector machines, regularization, optimization, and beyond. *Am. Stat. Assoc.* **98**(462), 489 (2002)
11. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: *IEEE International Conference on Computer Vision 2013*, pp. 854–863. IEEE, Sydney (2013)