International Symposium on Communication Engineering & Computer Science (CECS 2018)

Residual Network Based on Multi-Features Combination for Tracking

Qian Zou [a], Shaofu Lin [b] and Yanan Du[c]

School of Software Engineering, Beijing University of Technology, Beijing100124, China

[a]zouqian@emails.bjut.edu.cn, [b]linshaofu@bjut.edu.cn, [c]duyanan56@emails.bjut.edu.cn

Abstract. Correlation filter (CF) based tracking algorithms have shown favorable performance in recent years and have the impressive performance on benchmark datasets. The combination of deep learning and correlation filtering has also become a research hotspot. However, the tracking model has limited information about their context and easily drift in cases of fast motion, occlusion or background clutter, and the trackers update tracking models at each frame without considering whether the detection is accurate or not. In this paper, we present a tracking strategy based on the multi-features combination and use the residual network to enhance the learning ability that makes our trackers can take full advantage of multi-features. Experimental results on the benchmark datasets show that the performance of the model has been improved effectively.

Keywords: Multi-Features, Residual Learning, Correlation filtering, Visual Tracking.

1. Introduction

Visual tracking means to detect candidate regions by analyzing the image sequences of video, and then to locate the coordinates of these targets in the video. Object tracking is of great academic and practical importance to such fields as virtual reality (VR), human-computer interaction, monitoring system, augmented reality (AR) and machine perception and remains a core problem in computer vision. Large new datasets and benchmarks such as OTB-2013 [1], OTB-2015 [2], TempleColor128 [3], ALOV300++ [4] and UAV123 [5], as well as, tracking challenges such as the visual object tracking (VOT) challenge and multi-object tracking (MOT) challenge have attracted many researchers to make contribution to the development of this area.

The tracking problem can be divided into two main challenges: object representation and sampling for detection. Recently, many successful tracking algorithms use strong hand-crafted features, such as Histograms of Oriented Gradients (HOG) and Color names [6] or learned ones to represent the tracked object. Many deep learning researchers also use deep features trained on a large dataset, such as ImageNet, to represent the tracked object. On the other hand, sampling is needed to strike a balance between computation time and precise scanning of the region of interest for the target. Many researchers have focused on the correlation filter trackers, due to their high accuracy when they are running at high frame rates. In general, CF trackers learn a correlation filter online to localize the object in consecutive frames. The learned filter is applied to the region of interest in the next frame and the location of the maximum response corresponds to the object location. Then the filter is updated by using the new object location. The major reasons behind the success of this tracking paradigm are the approximate dense sampling performed by circularly shifting the training samples and the computational efficiency of learning the correlation filter in the Fourier domain. Staple tracker [7] is based on CF framework and combines HOG and color histogram feature to represent the object. The color histogram is very robust to deformation, but not to illumination, and HOG is robust to illumination. The Staple model designed two channels for learning the two features that complement each other and showed good performance. However, Staple based on CF have limited context information and easily drift in cases of fast motion, occlusion or background clutter. In addition, the tracker's update model at each frame without being considered whether the detection is accurate or not. As a result, target tracking may easily fail.

With the great power in the feature representations, CNNs have been demonstrated significant success on many computer vision natural language processing and speech recognition tasks, including visual tracking. In particular, the deep network could extract deep features to characterize targets in image processing. Under the large number of images of ImageNet, the ability of target recognition based on deep learning has exceeded the human recognition. However, in visual tracking, there are few positive and negative samples to train deep network and extract the strong features, which limiting DL application in visual tracking. Some researchers use so much unrelated data to train network to obtain the general expression ability, and then fine tune the network, then the extraction is better than HOG. Some researchers have also designed a simple neural network to speed up training process and proposed the tracker's network.

In this paper, we use CNN and histogram features to representing the object, so the residual network can take the place of CF models to output response map directly. We combine the response map obtained from the network and the response map obtained from the color histogram model to calculate the possible position of the target and make the tracker more robust.

2. Related Work

2.1 CF Framework.

In recent years, with the development of the correlation filter model in the field of visual tracking applications, many researchers have been interested in studying the high accuracy of this model while it is running at high frame rates. The correlation filter tracker trains the filter by inputting the object Gauss distribution representation feature and then searching the response peak in the prediction distribution to locate the target in the following tracking. The correlation filter uses the fast Fourier transform to obtain the large speed enhancement, and it has good performance on real-time, which is an important indicator of the tracker. A large number of optimized models based on CF are proposed, such as MOSSE, CSK, KCF, STC, CN, Staple and CACF [8, 9, 10, 11, 6, 7, 12]. MOSSE is a single-channel gray feature CF. CSK extends dense sampling (add padding) and kernel-trick based on MOSSE. KCF applies the multi-channel gradient HOG feature. STC interprets CSK model in a Bayesian framework and adds context information. CN mainly uses the multi-channel Colornames feature. CACF combines HOG features and contextual information based on the CF framework. Other models also use the histogram, such as DSST [13] and Staple. From above we can conclude that more and more optimization models fuse various kinds of features and make full use of the context of the object to enhance the robustness of the tracker and to reduce the influence of occlusion, motion blur, and illumination changes. CF trackers have achieved good results on benchmark datasets, such as OTB and VOT. Compared to some traditional tracking algorithms, KCF, Staple and other models have better performance and stronger robustness in applications.

2.2 DL Framework.

The object characteristic is very important for tracking. Deep learning can extract the features of the image in many dimensions, such as texture, structural and semantic features. Through this approach, the content of the image can be better understood by the model. When deep learning widely used in image recognition, many researchers have been considering how to apply it to track tasks. Wang [14] proposed a strategy based on deep feature extraction that mainly uses a large number of data to train the network to obtain the targets general expression. In special scenes, the network is fine tune by the object, then the appropriate tracking strategy is selected to track the object. Some researchers also used VGG-Net [15] that was trained on ImageNet to extract target features directly, which train CF to track the object. Hyeonseob [16] has proposed that use of video sequence data to train deep neural networks model. It breaks through the problem of less training samples and uses the context information in the sequence data better, but it is not sufficient to express the target features. And many researchers have proposed trackers based on Siamese-CNN to solve the tracking problem by combining the traditional methods such as the correlation filter or particle filter, and the performance is gradually improved. Besides, there is investigation on the recurrent neural network (RNN) to facilitate tracking as object verification. The feature extracted by CNN layers and RNN can put forward the time-series characteristics in sequence data [17], which will prevent the object drift very well.

3. Method

3.1 Residual Learning.

With the development of CNN, researchers can avoid the gradient disappearance by standard initialization and intermediate standard layer, and then stack more network layers and train a large number of data to improve the accuracy of the network. With the increase of network layers, the network becomes more complex. When the network converges to local minimum points, the accuracy will s saturation and decline rapidly. This decline is not caused by overfitting and when the network layers continue to increase that training error will rise rapidly. This degradation indicates that this system is not easy to optimize. He [18] proposed the concept of residual learning method, which reduces the influence of network performance degradation, and proposed a deep residual network. When a new layer is added to fit the optimal mapping. Rather than stacking more layers to approximate H(X), we expect these layers to approximate the residual function: F(X) = H(X) - X. Then the original fitting target is F(X) + X. Moreover, the nonlinear layer can be designed as a forward neural network with Shortcut. The residual block can write as H(X) = F(X) + X, and it can solve the degradation problem well that make the network layer deeper. The residual network is easily optimization and the result are actually better than the previous network.

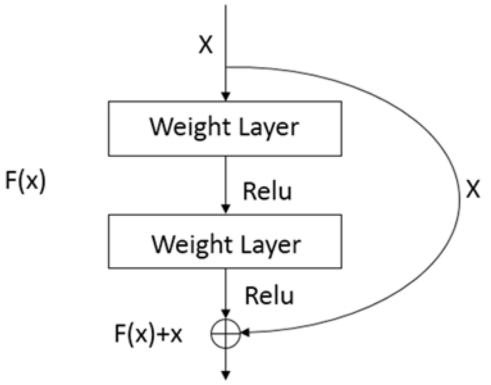

Fig 1. Residual Learning

Some trackers are designed by deep feature and CF which is based on CNN. The trackers improve the real-time performance, although it takes a lot of time to extract the deep feature that is more acceptable than training. The CF uses fast Fourier transform and kernel tricks to calculate the feature in Fourier domain so that the whole process is accelerated and simplified, and CF has strong real-time performance. In our trackers, we use residual learning to instead of the function of the CF, which can simplify the structure of the network and improve performance.

3.2 Network Architecture Design.

3.2.1 Network Design Strategy

The CF learn a filter and predict the target through searching maximum value position in the response map that is CF's output. In general, the linear correlation filter W is learned by solving the following minimization problem, X denotes input samples and Y denotes the corresponding Gaussian function label.

$$W^* = \arg\min_{w} \| W * X - Y \|^2 + \lambda \|W\|^2 \tag{1}$$

The convolution operation between the W and X is a dot product in Fourier domain, we regard the linear CF's operation as the loss minimization process of the CNN, then the form of the loss function can be written as:

$$\mathrm{L(W)} = \frac{1}{\mathrm{N}} \sum \mathrm{L_w(X^{(i)})} + \lambda \mathrm{r(W)} \tag{2}$$

Here, N is the number of samples, Lw(X(i)) is loss value of the i-th sample and r(W) is weight decay. We set N == 1 and take L2 norm as r(W), the loss function can be written as:

$$L(W) = L_W(X) + \lambda \| W \|^2 \tag{3}$$

When Lw(X) = || F(X) - Y ||, this function is equivalent to take L2 norm to compute loss value, where F(X) is the networks output and Y is the label. F=W*X is convolution operation compute by one convolution layer that is equivalent to the correlation filter, so, the network weights can be effectively calculated using the gradient descent and that simplified calculation.

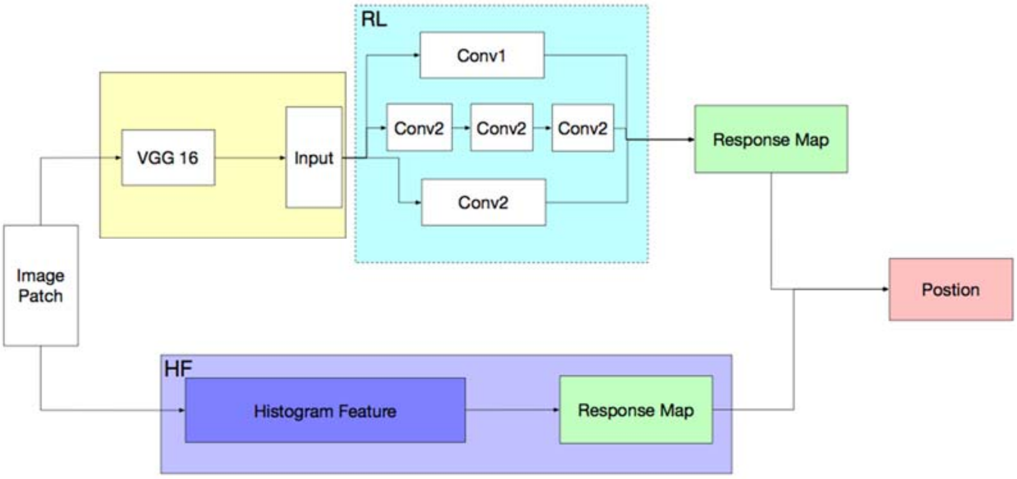

Fig 2. Architecture

When the convolution layer is used to replace the correlation filter, we can combine the residual network method to optimize the network structure. We denote F(x) is the optimal mapping of input X and FB(X) is the output from the base layer, we expect that residual layers to approximate FR(X). As a result, the output of convolution layer is close to the ideal target as possible and it could speed up the training process.

$$F(X) = F_B\left(X, \{W_B\}\right) + F_R\left(X, \{W_R\}\right) \tag{4}$$

When capturing the residual to fitting loss value, we design two branches to compute the temporal residual and the spatial residual and get more feature information from a different domain. The spatial residual is initializing and learning by every image patches that can reduce the influences from appearance changes. The temporal input is the first frame which contains the initial object appearance. So, the expression can be written as:

$$F(X_t) = F_R(X_t) + F_{SR}(X_t) + F_{TR}(X_1) \tag{5}$$

In the whole network, we use two channels to extract deep features and color histogram features, then train residual network and color histogram filter parallelly. When getting the new input frame, we first crop the search patch centered at the estimated position of the previous frame. The patch is sent to our feature extraction partition and then regressed into two response maps through the base and residual layers and histogram filter. Finally, the response maps will be merged accord to different weights and search the maximum value position in the response map as the object the last position. Actually, this tracker combines the characteristics of color histogram and deep CNN feature.

3.2.2 Tracking by Multi-Features

For deep features, we adopt the VGG-19 network for feature extraction which pre-trained on ImageNet dataset for object recognition, and obtained more general expression of the target in mass image data. Compared with HOG feature, deep feature reduce the real-time but improve the accuracy. And the histogram score is computed from an M-channel feature image , obtained from and defined on a (different) finite grid .

$$f_{hist}(x;\beta) = \frac{1}{|\mathrm{H}|} \sum_{u \in \mathrm{H}} \zeta_{(\beta,\psi)}[u] \tag{6}$$

The model uses two channel to learn residual network and histogram features so that the tracker can learn more context information trained by image patches. And then the model should get better performance. Normally the response map can merge with factors or use a Gaussian method, we choose set factors to adjust the tracking performance.

3.2.3 Scale Estimation

When we obtain the target center location, we extract search patches in different scales and then resized to the size of training patches. We choose the maximum corresponding scale to smooth scale estimation update. The width wt and height ht of the target object at frame t is updated as:

$$(\mathrm{w}_t, h_t) = \beta(\mathrm{w}_t^*, h_t^*) + (1-\beta)(w_{t-1}, h_{t-1}) \tag{7}$$

Where wt* and ht* are the width and height of the scaled object with maximum response value, the β is a smooth factor, and (wt, ht) denote patches size of the next frame.

3.3 Model Update

A tracker depends on the oscillation of response map to update model. Moreover, the tracker avoids error updates when the highest point Fmax is still high and has strong oscillation on response map. The first one is the maximum response score Fmax of the response map F(s,y;w) defined as:

$$F_{\max} = \max F(s, y; w) \tag{8}$$

We use the average peak-to-correlation energy method to measure the oscillation of the response map and defined as:

$$E = \frac{|F_{\max} - F_{\min}|^2}{mean(\sum_{w,h}(F_{w,h} - F_{\min})^2)} \tag{9}$$

Where Fmax, Fmin and Fw,h denote the maximum, minimum and the w-th row h-th column elements of F(s,y;w). This criterion can reflect the oscillation of the response map. In addition, E relative to the historical mean will be reduced obviously when the object is occluding or missing. Therefore, we do not choose to update the model so as to avoid model drift. When E and become larger than historical means, the model will update the template score, and histogram score learning parameters is updating every frame. In this way, the model will reduce drift and updates numbers, to accelerate the tracking speed.

4. Experiments

We get the training image patches from the first frame and the object position with the ground truth size. And then we use Hann window to compute the window size and get patches from the first image. Our Histogram model initialize by this image patches, and feature extraction network is from VGG-19 with the first two pooling layers. We extract image features from the conv4-3 layer and change the channels to 64 through PCA method. And then we set factor to merge the two response maps which output from network and Histogram model, we set scale estimation factor is to 0.6 and update the whole trackers. Our experiments are performed on a sever with Xeon 2.10GHz CPU and Tesla P100 GPU with MatConvNet toolbox.

The experiments are conducted on OTB2013 standard benchmarks. The two datasets contain 50 sequences, respectively, and in this sequence, we mainly evaluate the model from the following indicators: Illumination Variation, Scale Variation, Occlusion, Deformation, Motion Blur, Fast Motion, In-Plane Rotation, Out-of-Plane Rotation, Out-of-View, Background Clutters and Low Resolution.

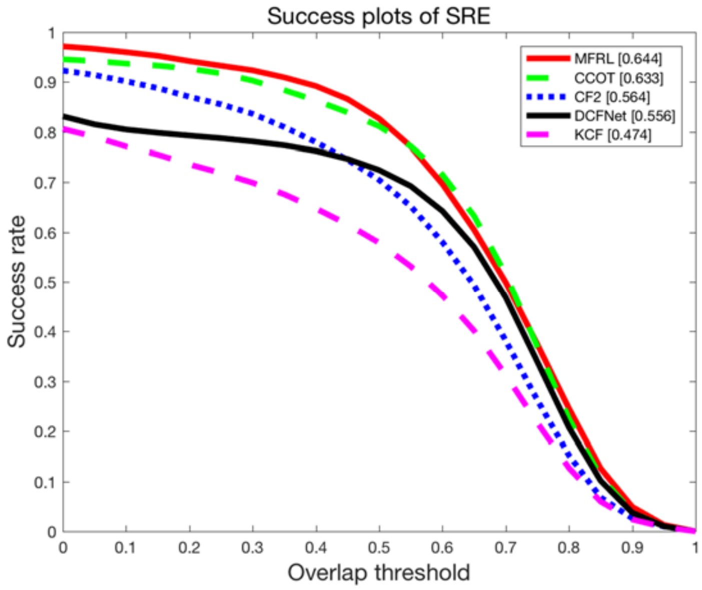

Fig 3. Success plots of SRE

We compered our tracking results with CCOT, CF2, KCF and DCFNet trackers, this is the stat-of-the-art tracker in object tracking. The Fig3 illustrates the success plots for SRE on OTB-13 benchmark, and the Fig4 illustrates the success plots for 6 challenging attributes. From the results, we can see that our model has showed better performance than stat-of-the-art trackers.

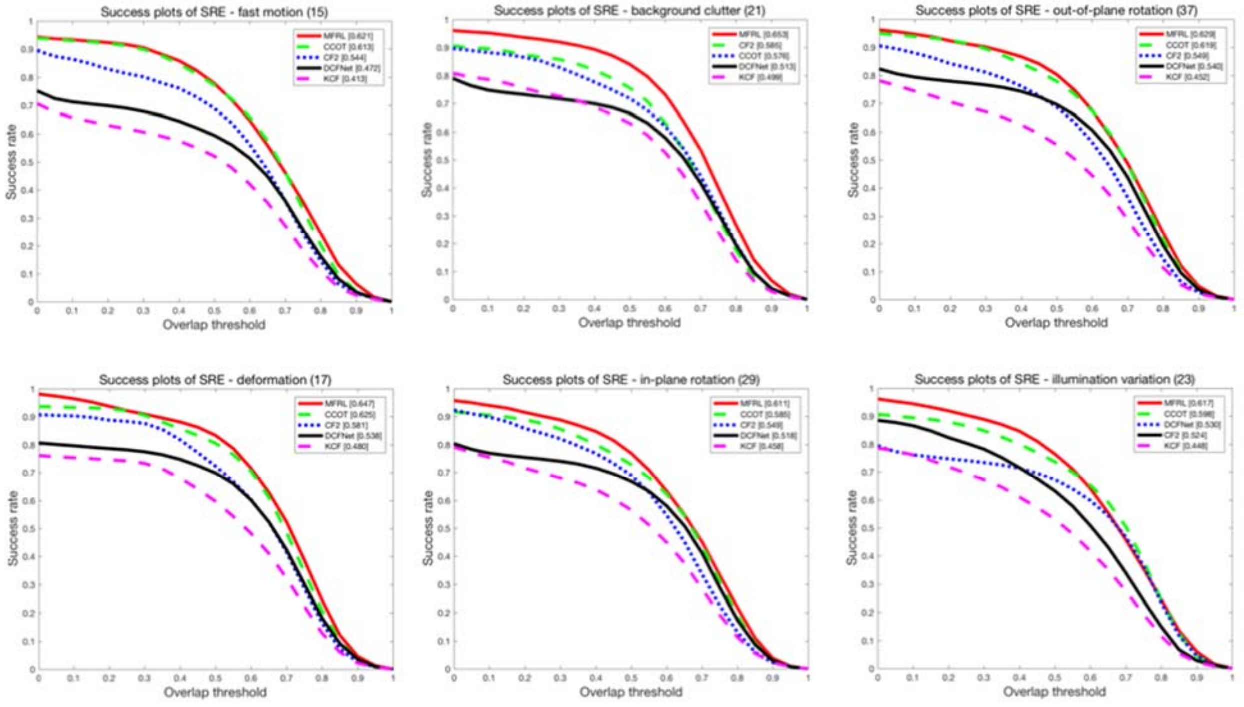

Fig 4. Success plots using SRE on the OTB-2013 dataset. MFRL is better than other state-of-the-art trackers

5. Conclusion

We used residual network to instead of correlation filter, and extract feature from two channels with CNNs and histogram. Meanwhile, compute the object position through merge the two response maps, and then the model is updated by observing the response graph without updating each frame. It not only reduces the computational complexity, but also improved the robustness of the model, and obtained more general expression of the target. The improved model has been tested on benchmark, and its performance has been effectively improved.

References

[1]. Wu Y, Lim J, Yang M H. Online Object Tracking: A Benchmark[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2013:2411-2418.

[2]. Wu Y, Lim J, Yang M H. Object Tracking Benchmark[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9):1834.

[3]. Liang P, Blasch E, Ling H. Encoding color information for visual tracking: Algorithms and benchmark[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2015, 24(12):5630.

[4]. Smeulders A W M, Chu D M, Cucchiara R, et al. Visual Tracking: An Experimental Survey[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(7):1442-1468.

[5]. Mueller M, Smith N, Ghanem B. A Benchmark and Simulator for UAV Tracking[J]. 2016.

[6]. Mueller M, Smith N, Ghanem B. Context-Aware Correlation Filter Tracking[C]// Conference on Computer Vision and Pattern Recognition. 2017.

[7]. Bertinetto L, Valmadre J, Golodetz S, et al. Staple: Complementary Learners for Real-Time Tracking[J]. 2015, 38(2):1401-1409.

[8]. Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters[C]// Computer Vision and Pattern Recognition. IEEE, 2010:2544-2550.

[9]. Rui C, Martins P, Batista J. Exploiting the circulant structure of tracking-by-detection with kernels[C]// European Conference on Computer Vision. Springer-Verlag, 2012:702-715.

[10]. Henriques J F, Rui C, Martins P, et al. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(3):583-596.

[11]. Zhang K, Zhang L, Yang M H, et al. Fast Tracking via Spatio-Temporal Context Learning[J]. Computer Science, 2013.

[12]. Mueller M, Smith N, Ghanem B. Context-Aware Correlation Filter Tracking[C]// Conference on Computer Vision and Pattern Recognition. 2017.

[13]. Danelljan M, Häger G, Khan F S, et al. Accurate Scale Estimation for Robust Visual Tracking[C]// British Machine Vision Conference. 2014:65.1-65.11.

[14]. Wang N, Yeung D Y. Learning a deep compact image representation for visual tracking[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2013:809-817.

[15]. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[16]. Nam H, Han B. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking[J]. 2015:4293-4302.

[17]. Fan H, Ling H. SANet: Structure-Aware Network for Visual Tracking[J]. 2016:2217-2224.

[18]. He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778.