A RANKING BASED ATTENTION APPROACH FOR VISUAL TRACKING

Shenhui Peng[⋆] *Sei-ichiro Kamata*[⋆] *Toby P. Breckon*[†]

[⋆] Graduate School of Information, Production and Systems, Waseda University, Japan
[†]Engineering and Computing Science, Durham University, UK

ABSTRACT

Correlation filters (CF) combined with pre-trained convolutional neural network (CNN) feature extractors have shown an admirable accuracy and speed in visual object tracking. However, existing CNN-CF based methods still suffer from the background interference and boundary effects, even when a cosine window is introduced. This paper proposes a ranking based or guided attention approach which can reduce background interference with only forward propagation. This ranking stores several convolution kernels and scores them. Subsequently, a convolutional Long Short Time Memory network (ConvLSTM) is used to update this ranking, which makes it more robust to the variation and occlusion. Moreover, a part-based multi-channel convolutional tracker is proposed to obtain the final response map. Our extensive experiments on established benchmark datasets show comparable performance against contemporary tracking approaches.

***Index Terms*—** Visual tracking, Ranking based attention, Convolutional tracker, ConvLSTM

1. INTRODUCTION

In recent years, visual tracking algorithms have evolved rapidly, as a fundamental and critical topic within computer vision, with various applications such as autopilot system, video surveillance and human-computer interaction interface. In visual tracking algorithms, the object categories should not be limited by the training set. Since the specific kind of target will not be known in advance of the tracking task, the visual tracking algorithm should be robust enough to track any kind of the object and can be promptly specialized according to the information obtained from the initial frame. Meanwhile, robust trackers should also deal with tough challenges such as appearance variations, motion blur, scale changes, illumination changes and occlusion.

Early tracking algorithms tended to use well-designed hand-craft feature extractors such as HOG [1] and SIFT [2]. At that time, correlation filter based tracking algorithms were widely considered due to their superior accuracy [3]. Kernelized correlation filters [4] offered a good way to speed up tracking by changing the convolution operation to an element-wise product via the Fourier domain. Spatially regularized correlation filters [5] were proposed to deal with boundary effects. In recent years, with the development of deep learning, convolutional neural networks (CNN) are widely used in the field of computer vision, such as image classification and semantic segmentation. Some studies [6] have shown that the feature maps from shallow layers in CNN contain object texture and localised detail information. In contrast, feature maps from deeper layers contain semantic information of the input image. As a result, many new tracking algorithms have been proposed based on CNN feature extractors [7] [8]. Further more, there exist some approaches which attempt to use convolution operation instead of correlation operation to achieve end-to-end training [8]. However, some papers [9] point out that existing deep learning based methods are unstable and time consuming with the use of stochastic gradient descent (SGD) to update the kernel on a frame by frame basis.

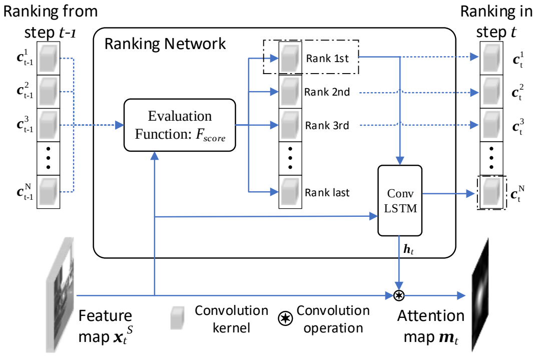


Fig. 1. Flow chart of the ranking network in time step t. Ranking will use feature x_t^S and function F_{score} to score the stored kernels and select the first and the last.

Although CNN have an admirable ability to extract features, it is still not enough to eliminate the influence of background interference only by the backpropagation of a loss function. This is why [8] proposed residual connections to enhance performance. In convolutional trackers, the convolution kernel should be the same size as object which will commonly result in network instability and training difficulties. The work of [9] has advantages in speed and memory saving, using convolutional Long Short Time Memory network (ConvLSTM). However there still exists some problems such

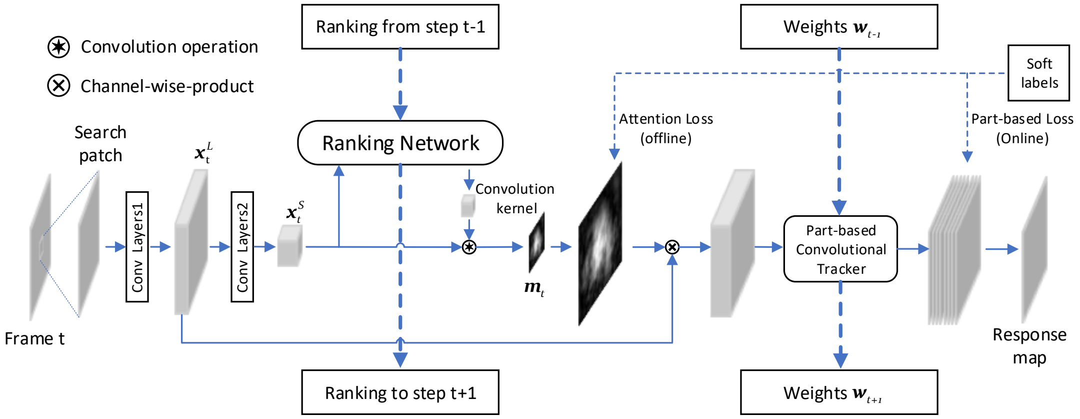


Fig. 2. Overall structure of the ranking based attention network. x_t^L and x_t^S are feature maps with different scale levels. m_t is the attention map as also shown in Fig. 3(b) .

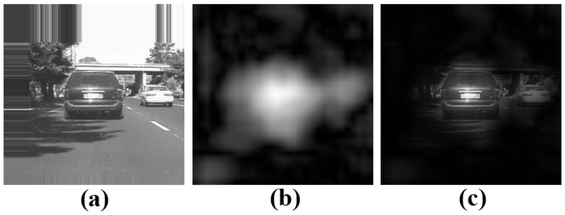


Fig. 3. Visualization of the feature map (a), replaced with the search patch, the attention map (b) and the results of attention map multiplied with feature map (c), which show that the background interference is significantly suppressed.

as low spatial resolution and lacking robustness to occlusion.

To address the aforementioned problems, we propose a ranking based attention network for visual tracking as shown in Fig. 1 and Fig. 2. This ranking stores several convolution kernels which contain semantic information of the target. If we perform convolution operations between these kernels with feature maps, we obtain response maps as shown in Fig. 3, which can significantly suppress background interference. The ranking network is initialized at the first frame. In each frame, the ranking re-scores and re-ranks the stored convolution kernels. The first one will be chosen as the output, the last one will be updated by using the ConvLSTM. We also propose a part-based multi-channel convolutional tracker which reduces the kernel size and improves robustness to occlusion. The main contributions of this paper are summarized as follows:

- We propose a ranking network to guide the usage and update of convolution kernels within an adaptive visual tracking framework. It can make the attention map stable, especially when occlusion occurs.
- We propose a part-based multi-channel convolutional tracker. It can speed up network updates and make the model more robust to occlusion.

2. THE PROPOSED METHOD

This section will describe our framework in detail. The ranking network is the key component, which will store several convolution kernels, score them and then pick the best one as the output. ConvLSTM is used to update these stored kernels. Within the part-based multi-channel convolutional tracker, a large convolution kernel is divided into several smaller kernels in multiple channels and then using a novel kernel to combine these channels together, which achieves the same effect of part-based tracking.

2.1. Ranking Based Attention

Firstly, the ranking network stores several convolutional kernels which contain the semantic information of the target. We obtain an attention map as shown in Fig. 3(b), after the convolutional operation between convolution kernel and feature map. Background interference and boundary effect can be significantly eliminated after we multiply the feature map with the attention map in channel-wise as shown in Fig. 3(c).

A ranking network is proposed to store and score several convolution kernels. In each frame or time step, the rank will score the stored kernels using the current feature map and choose the best one as the output. The output kernel will convolve with the feature map from *Conv Layers2* to get the attention map as shown in Fig. 2. The upsampled attention map will be multiplied with the feature map from *Conv Layers1* using a residual connection as shown in Fig. 2, which ensures the spatial resolution is guaranteed while background interference is reduced.

Specifically, the ranking is described as an **ordered set**, $\{c_t^1, c_t^2, \dots, c_t^N\}$, $c_t^i \in \mathbb{R}^{11\times11\times512}$, $i = 1, 2, \dots, N$, as shown in Fig. 1. In each time step or frame t, $x_t^S \in \mathbb{R}^{H/8\times W/8\times512}$ and $x_t^L \in \mathbb{R}^{H/4\times W/4\times256}$ are extracted from *Conv Layers2* and *Conv Layers1* respectively as shown in

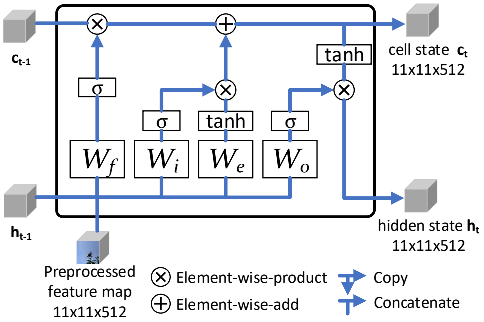


Fig. 4. Flow chart of convolutional LSTM. As stated in [9], σ and $tanh$ correspond to $sigmoid()$ and $tanh()$ activation functions respectively. W_f, W_i, W_e and W_o are the convolutional filters weights for the forget gate, input gate, estimated cell gate and output gate.

Fig. 2. H and W are the height and width of the search patch which is input into the feature extraction network. The rank will score the stored kernels using a score function F_{score}:

$$s_t^i = F_{score}(\boldsymbol{c}_t^i, \boldsymbol{x}_t^S) = PNR(\boldsymbol{c}_t^i * \boldsymbol{x}_t^S),\ s^i \in \mathbb{R}, \tag{1}$$

where $*$ represents the convolution operation. We use the same method in [10] to calculate the PNR score:

$$PNR(\boldsymbol{m}) = \frac{max(\boldsymbol{m}) - min(\boldsymbol{m})}{mean(\boldsymbol{m}/max(\boldsymbol{m}))},\ \boldsymbol{m} \in \mathbb{R}^{H/8 \times W/8}, \tag{2}$$

of response map for each $\boldsymbol{c}_t^i$. The ordered set is re-ordered by the scores:

$$\{\boldsymbol{c}_t^1, \boldsymbol{c}_t^2, \ldots, \boldsymbol{c}_t^N\} \Leftarrow ordering\{s_t^1, s_t^2, \ldots, s_t^N\}. \tag{3}$$

The $\boldsymbol{c}_t^{(1st)}$ with the top score will be selected as the output of the rank. The $\boldsymbol{c}_t^{(Nth)}$ with the lowest score will be updated:

$$\boldsymbol{c}_t^{(1st)} = \mathop{\arg\max}_{\boldsymbol{c}_t^i}\{s_t^i \mid i = 1, 2, \ldots, N\}, \tag{4}$$

$$\boldsymbol{c}_t^{(Nth)} = \mathop{\arg\min}_{\boldsymbol{c}_t^i}\{s^i \mid i = 1, 2, \ldots, N\}. \tag{5}$$

The ConvLSTM is used to update $\boldsymbol{c}_t^{(Nth)}$ and the hidden state $\boldsymbol{h}_{t-1}$:

$$(\boldsymbol{h}_t, \boldsymbol{c}_t^{(Nth)}) = ConvLSTM(\boldsymbol{h}_{t-1}, \boldsymbol{c}_t^{(1st)}, \boldsymbol{x}_t^{pre}). \tag{6}$$

We use $PreNet$, a pre-trained 2-layers CNN, to downsample the $\boldsymbol{x}_t^S$ and match the size of $\boldsymbol{h}_{t-1}$:

$$\boldsymbol{x}_t^{pre} = PreNet(\boldsymbol{c}_t^{(1st)}, \boldsymbol{x}_t^S),\ \boldsymbol{x}_t^{pre} \in \mathbb{R}^{11 \times 11 \times 512}. \tag{7}$$

Finally, the attention map $\boldsymbol{m}_t$ can be generated from the convolution operation between the $\boldsymbol{h}_t$ and feature map $\boldsymbol{x}_t^S$:

$$\boldsymbol{m}_t = \boldsymbol{h}_t * \boldsymbol{x}_t^S,\ \boldsymbol{m}_t \in \mathbb{R}^{H/8 \times W/8}, \tag{8}$$

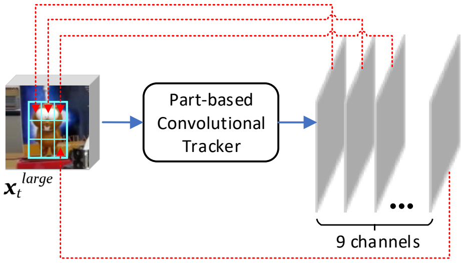


Fig. 5. Each output channel is concentrated on one part of tracking the target.

as shown in Fig. 3(b) for visualization.

In offline training stage, the loss function $\mathcal{L}(\boldsymbol{h}_t)$:

$$\mathcal{L}(\boldsymbol{h}_t) = \frac{1}{\frac{H \times W}{8 \times 8}} \sum_{h,w} (\boldsymbol{h}_t * \boldsymbol{x}_t^S - \boldsymbol{y}_t^S)^2 + \lambda_1 \|\boldsymbol{h}_t\|^2, \tag{9}$$

is proposed to push the ConLSTM to learn how to update convolution kernels , where $\boldsymbol{y}_t^S \in \mathbb{R}^{H/8 \times W/8}$ is a two-dimensional gaussian label centered on the target position for each search patch and λ_1 is the regularization parameter. The ranking network will degrade to normal ConvLSTM since the ranking length N will be set as 1, which makes the gradient back propagation normally.

2.2. Part-based Multi-channel Convolutional Tracker

Following closely, the feature map $\boldsymbol{x}_t^L \in \mathbb{R}^{H/4 \times W/4 \times 256}$ is multiplied with the upsampled attention map in channel wise, which will be input into part-based multi-channel convolutional tracker. The target will be split into 9 parts by a 3 × 3 grid. There are 9 channels in this tracker. Each channel will concentrate on tracking one part of the target, as illustrated in Fig. 5. The tracker is described as the convolution operation between weight $\boldsymbol{w}_t \in \mathbb{R}^{9 \times 7 \times 7 \times 256}$ and feature $\boldsymbol{x}_t^L$, as stated in [8]. The output of the tracker contains 9 channels. Each channel has a 2D gaussian distribution like response map and the peak of the response map represents the position of the corresponding part. A convolution kernel with prior information is proposed to generate the final one channel response map, which combines the 9 channels together. The position of the whole target is the peak position of the final response map.

In the online updating stage, the online training dataset will be collected from last T frames. The last T tracking results will be used to generate the training labels $\boldsymbol{y}_t^L \in \mathbb{R}^{9 \times H/4 \times W/4}$. The loss function is described as:

$$\mathcal{L}(\boldsymbol{w}_t) = \frac{1}{\frac{H \times W}{4 \times 4}} \sum_{l=1}^{9} \sum_{h,w} (\boldsymbol{w}_t^{(l)} * \boldsymbol{x}_t^L - \boldsymbol{y}_t^{L(l)})^2 + \lambda_2 \|\boldsymbol{w}_t\|^2, \tag{10}$$

where λ_2 is the regularization parameter.

3. EXPERIMENT

In this section, we first explain the implementation details. Subsequently, we compare our tracker with state-of-the-art trackers on OTB2013 [11] and OTB2015 [12] datasets. Ablation studies are adopted to evaluate each component.

3.1. Implementation Details

The hardware environment is a workstation with i7-6800K CPU, 16GB RAM and GTX1080Ti GPU.

Offline training stage: In this stage, only ranking network is trained. The training datasets are UAV123 [13] and TC128 [14]. The overlap sequences with the test set [11] [12] are eliminated in advance. We generate 2D Gaussian labels $\boldsymbol{y}^S$ for every search patch in each frame. We use the patch-label pairs to train the ConvLSTM with the loss function (9). The ranking network degrades to normal ConvLSTM with $N = 1$ to make the gradient back propagation normally.

Online tracking stage: In this stage, all of the parameters in ranking network are fixed. We just update the parameters in the part-based convolutional tracker. The last T frames are collected as the online training set. The multi-channel labels $\boldsymbol{y}^L$ are generated based the last T tracking results. The loss function Eqn.(10) is used to update the weights. As for the feature extracter, we use a pre-trained VGG16 [15] network. We generate a batch of search patches with different scales and find the best response to do scale estimation. We use a simple initialization network to initialize the ranking network, $\boldsymbol{h}_0$ and $\boldsymbol{w}_0$ at the first frame. The results below are from the following super-parameter settings, $N = 3$, $T = 12$, $H = W = 255$, $\lambda_1 = 1e - 10$ and $\lambda_1 = 1e - 7$.

3.2. Overall Performance

We select several state-of-the-art trackers ACT [16], CREST [8], metaCREST [17], SRDCF [5], siamPRN [18], siamFC3s [19] and staple [20]. Fig. 6 illustrates the results in OTB2013 [11] and OTB2015 [12] datasets. In OTB2013, our tracker ranks the second in terms of precision and in OTB2015 ranks the first. The other trackers precision drops rapidly in OTB2015, since its sequences number is two times larger than OTB2013 and contains several challenging sequences. Our tracker is robust enough to deal with such problems.

3.3. Ablation Studies

In order to evaluate the contribution of each component, ablation studies are conducted. We construct a tracker without ranking network as the baseline. It indicates that our part-based convolution tracker is robust, especially in OTB2015 dataset as shown in Fig. 7. The proposed ranking network can lead the precision much better than the baseline, since the background interference is significantly suppressed by the ranking based attention map.

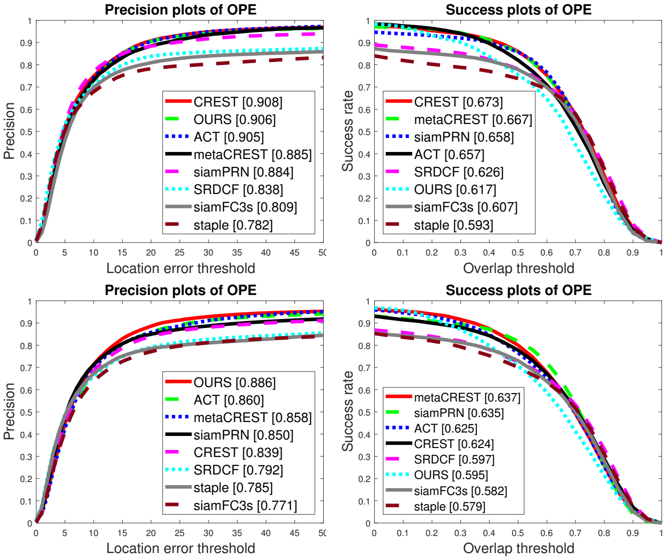
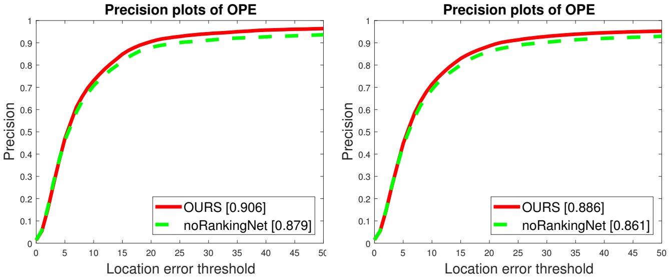


Fig. 6. Precision and overlap results. First row is the results of OTB2013, second row is the results of OTB2015.



(a) OTB2013 dataset (b) OTB2015 dataset

Fig. 7. The results of ablation studies.

4. CONCLUSION

In this paper, we present a novel ranking based attention network, which consists of an ranking network and a part-based multi-channel convolutional tracker. Our ranking network can significantly reduce the influence of background interference and uses ConvLSTM to update attention kernels. The part-based multi-channel convolutional tracker is shown to make the model more robust to occlusion. We also compare our tracker with contemporary tracking approaches, and obtain comparable results. Ablation studies show the effectiveness of each component.

5. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 18K11380.

6. REFERENCES

[1] Navneet Dalal and Bill Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition*. IEEE, pp. 886–893.

[2] David G Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui, “Visual object tracking using adaptive correlation filters,” in *Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2544–2550.

[4] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[5] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, “Learning spatially regularized correlation filters for visual tracking,” in *International Conference on Computer Vision*. IEEE, 2015, pp. 4310–4318.

[6] Matthew D Zeiler and Rob Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[7] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, “Convolutional features for correlation filter based visual tracking,” in *International Conference on Computer Vision Workshops*. IEEE, 2015, pp. 58–66.

[8] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson Lau, and Ming-Hsuan Yang, “Crest: Convolutional residual learning for visual tracking,” in *International Conference on Computer Vision*. IEEE, 2017, pp. 2555–2564.

[9] Tianyu Yang and Antoni B Chan, “Recurrent filter learning for visual tracking,” in *International Conference on Computer Vision Workshops*. IEEE, 2017, pp. 2010–2019.

[10] Zheng Zhu, Guan Huang, Wei Zou, Dalong Du, and Chang Huang, “Uct: learning unified convolutional networks for real-time visual tracking,” in *International Conference on Computer Vision Workshops*. IEEE, 2017, pp. 1973–1982.

[11] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Online object tracking: A benchmark,” in *Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 2411–2418.

[12] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Object tracking benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[13] Matthias Mueller, Neil Smith, and Bernard Ghanem, “A benchmark and simulator for uav tracking,” in *European Conference on Computer Vision*. Springer, 2016, pp. 445–461.

[14] Pengpeng Liang, Erik Blasch, and Haibin Ling, “Encoding color information for visual tracking: Algorithms and benchmark,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

[15] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015, pp. 1–14.

[16] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu, “Real-time ’actor-critic’ tracking,” in *European Conference on Computer Vision*. Springer, 2018, vol. 11211, pp. 328–345.

[17] Eunbyung Park and Alexander C. Berg, “Meta-tracker: Fast and robust online adaptation for visual object trackers,” in *European Conference on Computer Vision*. Springer, 2018, vol. 11207, pp. 587–604.

[18] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu, “High performance visual tracking with siamese region proposal network,” in *Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 8971–8980.

[19] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, “Fully-convolutional siamese networks for object tracking,” in *European Conference on Computer Vision*. Springer, 2016, pp. 850–865.

[20] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr, “Staple: Complementary learners for real-time tracking,” in *Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 1401–1409.