ROBUST VISUAL TRACKING WITH DEEP FEATURE FUSION

*Guokun Wang, Jingjing Wang, Wenyi Tang, Nenghai Yu**

CAS Key Laboratory of Electromagnetic Space Information,
University of Science and Technology of China, Hefei, China

ABSTRACT

Recently, CNN (Convolutional Neural Network) based trackers have achieved promising results benefited from their robust feature representation. However, most trackers only use features from a certain layer, which limits their performance. In this paper, we propose a novel CNN based tracker. Firstly, we use local detection and global detection network for target localization. In local detection network , we fuse features from different layers to train a fully convolutional neural network for target localization. In case the local detection network fails when the target disappear for a while and appears in another location, we train a global detection network to detect if the target appears again. Then, we employ a correlation filter to estimate accurate scale of the target using HOG features extracted around predicted location. Extensive experiments on various challenging video sequences demonstrate the effectiveness of our proposed algorithm compared with several state-of-the-art trackers.

Index Terms— Visual Tracking, feature fusion, scale estimation, Convolutional Neural Network

1. INTRODUCTION

Visual tracking has been playing a critical role in computer vision for a long time which aims to estimate the bounding box of a given target in each frame of an image sequence. Although it has been studied for decades, visual tracking is still an unsolved problem , because it requires designed trackers to be robust enough to handle several challenging factors such as occlusion, deformation and fast motion, which lead to large appearance change.

Many early-stage trackers rely on hand-crafted features to describe the target. Although they may overcome some challenging factors, the poor generalization ability limits their performance in visual tracking.

Recently, Convolutional Neural Networks (CNNs) have demonstrated breaking performance in computer vision tasks such as classification and detection [1, 2, 3, 4] due to its powerful capability of learning feature representation. Several CNN based trackers [5, 6, 7, 8, 9] are proposed, and their performance has surpassed traditional methods which rely on hand-crafted features to describe the target.

Fig. 1. Tracking results using CNN features from different convolutional layers on a representative frame of two sequences with various challenging factors. (a) VGG conv4-3. (b) VGG conv5-3. (c) Ground truth.

But there are several limitations of using CNN features for tracking. Most trackers only use feature maps from the last layer, which carry rich high-level semantic information. Although these features are strong at distinguishing the target from other types of objects, discarding spatial details of the tracked target make it diffcult to distinguish different objects from the same category. Features from lower layer which keep more details are more powerful to distinguish objects from the same category. However it is not robust against various challenging factors, as shown in Figure 1. It is imperative to combine features from different layers together to achieve better tracking performance,

In this paper, we propose a novel CNN based tracker, which can obtain precise location and accurate scale estimation. At first, we apply VGG-Net [1] to extract hierarchical convolutional feature maps from a rectangle area around the predicted position of the previous frame. Then, we use skip layer fusion to combine features from high-level (conv5-3) and low-level (conv4-3) together to predict target location precisely via local deteciton network. If the target is occluded by other objects, and then appears in another location, our local detection network may fail to follow it. To solve this problem we train a global detection network to search for a possible position through the whole image to detect if it appears again. After we get the target location, we estimate scale by applying correlation filter on the HOG features extracted from the area around that location.

The contributions of this paper are summarized below:

i) We develop an algorithm that uses global detection net-

*Corresponding author: ynh@ustc.edu.cn. This work is supported by NSFC (No: 61371192, 61472379).

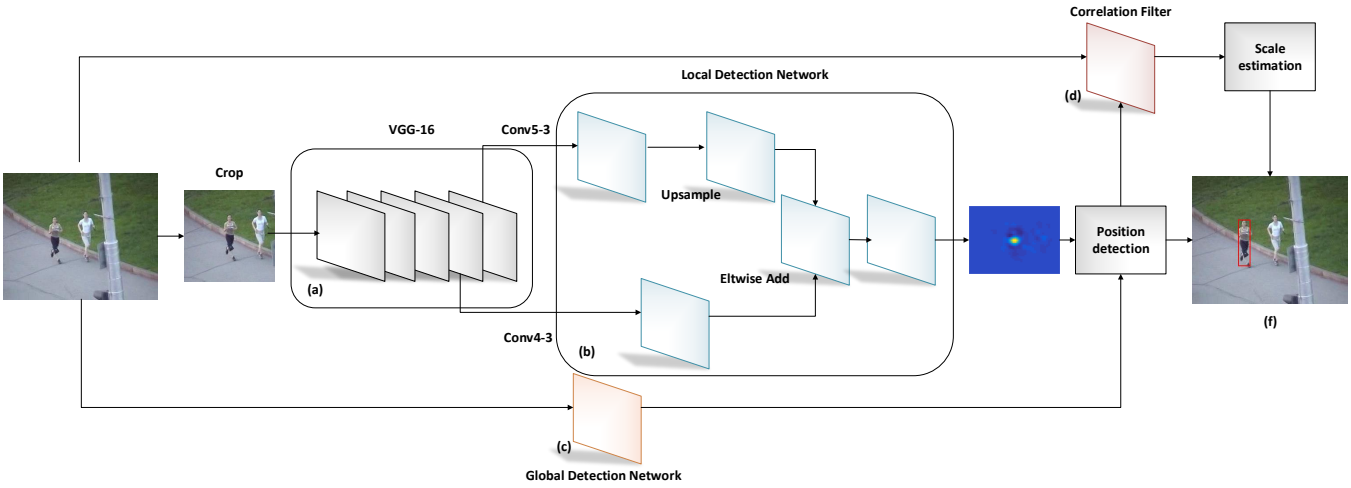

Fig. 2. The framework of our proposed tracking algorithm, (a) Feature extraction network. (b) Local detection network. (c) Global detection network. (d) Scale estimation correlation filter.

work and local detection network working cooperatively for target localization.

ii) We propose a novel tracking algorithm that combines feature maps from different layers to train local detection network.

iii) We conduct extensive experiments on 12 challenging video sequences and the result demonstrates that the proposed tracking algorithm outperforms several state-of-the-art trackers.

2. PROPOSED METHOD

As shown in Figure 2, our algorithm consists of four modules. At first, we extract CNN features from the region of interest (ROI) in current frame. Then, we use local detection and global detection network to determine the target location. Next, we use HOG features extracted around that location to estimate the scale. Finally, we get a credible bounding box of the tracked target.

2.1. Feature Extraction Network

We use the VGG-Net (16 layers) trained on Image Classification task [10] as the feature extraction network because it can provide richer information. Since visual tracking needs to localize the target precisely, we not only use high-level features extracted from conv5-3, but also use that from conv4-3 for more details.

2.2. Local Detection Network

As declared in [9], FCNT bulids two networks named SNet and the GNet on top of the selected conv4-3 and conv5-3 feature maps, and proposes an algorithm to decide when to use SNet or GNet for detection. But in his method, he uses target location predicted by GNet as default, which is based on the conv5-3 layer. It may lead to larger center error because the high-level features are more category-level and coarse.

When our proposed method makes a prediction of target location, we use a skip layer fusion to integrate more precise low-level features with high-level features together. Firstly, a deconvolution layer is used to upsample the feature maps come from conv5-3 via bilinear interpolation, followed by a crop layer which ensures its size is the same as that of conv4-3 layer's feature maps, then they are added together. At last, we use a convolutional layer which has convolutional kernels of size 5×5 and outputs the predicted heat map of input image.

Local detection network is initialized in the first frame, we crop a rectangle image region I_1 centered at ground truth with twice the size of the target bounding box, and reshape it in a fix size. Then we propagate forward the reshaped ROI through feature extraction network to get corresponding feature maps. We train local detection network by minimizing the following loss function:

$$L_{local} = \left\| \widehat{M}_1 - M_1 \right\|_2^2 \tag{1}$$

Where $\widehat{M}_1$ represents the foreground heat map predicted by the network; M_1 is a Gaussian distribution centered at the ground truth target location.

In frame t, we crop a rectangle image I_t centered at previous predicted location and pass it through feature extraction network. Then we feed feature maps from conv4-3 and conv 5-3 into local detection network and get a predicted heat map. The location with the maximum value on the heat map is selected as the center location of the target. The maximum heat map value then serves as the confidence $conf_t$ of this prediction. If the confidence is lower than a threshold, we think the

target is lost. Then we use global detection network to help search for a possible position over the whole image.

Online update. In order to adapt to the appearance change during the tracking process, we need to update our local detection network. To avoid updating using contaminated training samples, online update is conducted only if the confidence of the location prediction is higher than a predefined threshold θ. During online update, we also use information of the first frame to improve the discriminative power for foreground and background. Overall, local detection network is updated by minimizing:

$$L_{local} = \left\| \widehat{M_1} - M_1 \right\|_2^2 + \left\| \widehat{M_t} - M_t \right\|_2^2 \tag{2}$$

Where $\widehat{M_t}$ represents the foreground heat map predicted by the network; M_t is a Gaussian distribution centered at the tracking result in previous frame.

2.3. Global Detection Network

In case the local detection network loses the target, it is inefficient to move local detection network through the whole image to find a possible location. We turn to RPN (Region Proposal Network), because it takes an image as input and outputs a set of rectangular object proposals, each with an objectess score, as proposed in [3]. Thus, if the target is lost, we just put the next frame into RPN. Once the highest proposal score exceeded a threshold, we think the target is found, and mark the center of that proposal as the target location. In the next frame, we use local detection network to track the target again based on this location.

RPN slides a small network over the convolutional feature maps which is conv4-3 in our algorithm. This small network takes as input a 3×3 spatial window over the feature maps. Next, its output is mapped to a lower-dimensional feature and then fed into two sibling fully-connected layers: a box-regression layer (reg) and a box-classification layer (cls) to generate region proposals.

In the first frame, we train RPN end-to-end by back-propagation and SGD, and fix its parameters during tracking to avoid the distraction introduced by online update. The training process follows [3], but there are two differences:

1) For visual tracking, we just need to distinguish foreground from background, so we set the number of classes to 2, 0 for background and 1 for foreground.

2) The motivation of using RPN is to get proposals, so we do not use 4-step alternating training. Due to the limitation of training sample, we set the parameters of layers before conv5-3 fixed, and fine-tune RPN using the first frame and given ground truth.

2.4. Scale estimate correlation filter

During the tracking process, the target scale may change greatly. Besides an accurate localization, we need accurate scale estimation too. We use the method proposed at [11] to estimate scale change of the target using correlation filters with HOG features. That is , given the center location via local detection network or global detection network, we compute a pyramid in a rectangular area around the target location, and then set this pyramid to a rectangular cuboid with size of $M \times N \times S$, where M and N are the height and width of the filter and S is the number of scales. And then we extract its HOG features as train sample, calcualte desired correlation output g with a 3-dimensional Gaussian function. We initialize and update the scale space tracking filter using following equation:

$$A_t^l = (1-\eta)A_{t-1}^l + \eta \overline{G}_t F_t^l, \tag{3a}$$

$$B_t = (1-\eta)B_{t-1}^l + \eta \sum_{k=1}^{d} \overline{F_t^k} F_t^k \tag{3b}$$

G_t denote the FFT solution of desired correlation output of frame t, and F_t^l denote the FFT solution of l-channel HOG features of frame t, η is the learning rate, $A_1^l = \overline{G}_1 F_1^l$, $B_1 = \sum_{k=1}^{d} \overline{F_1^k} F_1^k$.

To estimate the scale of target in current frame, we extract a $M \times N \times S$ rectangular cuboid z as we mention above, and compute correlation scores y using

$$y = \mathscr{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B + \lambda} \right\} \tag{4}$$

The new scale is obtained by finding the maximum score in y.

3. EXPERIMENTS

3.1. Experimental Setting

The proposed tracker is implemented in Matlab with Caffe[12] framework. Both local detection network and global detection network are trained via online SGD with learning rate of 8e-7 and 0.025. We use 80 iterations to train the local detection network,100 iterations to train global detection network. When we start tracking, local detection network is finetuned for 2 iterations at each update step. The threshold θ for online update is set to 0.15. If the significance is less than 0.05, we use global detection network to search for possible position. If its highest score of proposal is larger than 0.4, we think that proposal is valid and use its center as new target position. All parameters are fixed throughout the experiments.

3.2. Experimental Results

We evaluate the performance of our proposed algorithm on 12 challenging video sequences compared with 8 state-of-the-art trackers including Struck [13], TLD [14], L1APG [15],

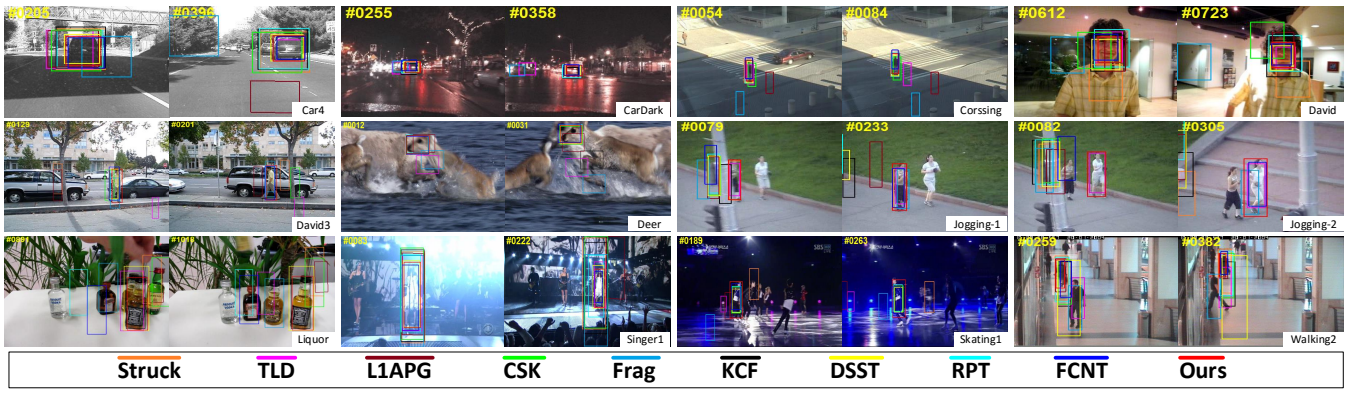

Fig. 3. Screenshots of tracking results of 10 trackers on 12 video sequences

Table 1. Average overlap rate. Red fonts indicate the best performance while blue fonts indicate the second best.

	Struck	TLD	L1APG	CSK	Frag	KCF	DSST	RPT	FCNT	Ours
car4	0.49	0.63	0.25	0.47	0.21	0.48	0.79	0.49	0.82	0.79
carDark	0.89	0.45	0.88	0.76	0.32	0.61	0.78	0.72	0.66	0.88
crossing	0.68	0.4	0.21	0.48	0.31	0.71	0.76	0.78	0.68	0.8
david	0.24	0.72	0.54	0.4	0.11	0.54	0.69	0.63	0.68	0.76
david3	0.29	0.1	0.29	0.5	0.67	0.77	0.46	0.8	0.74	0.78
deer	0.74	0.6	0.6	0.75	0.12	0.62	0.8	0.78	0.71	0.81
jogging-1	0.69	0.63	0.17	0.18	0.46	0.19	0.18	0.19	0.69	0.79
jogging-2	0.14	0.69	0.15	0.14	0.5	0.12	0.14	0.13	0.67	0.73
liquor	0.41	0.52	0.2	0.25	0.29	0.86	0.56	0.63	0.7	0.87
singer1	0.36	0.73	0.28	0.36	0.24	0.35	0.8	0.47	0.75	0.78
skating1	0.31	0.19	0.1	0.5	0.13	0.49	0.44	0.53	0.47	0.68
walking2	0.51	0.31	0.76	0.46	0.28	0.4	0.25	0.49	0.64	0.76
avg	0.48	0.5	0.37	0.44	0.3	0.51	0.56	0.55	0.68	0.79

Table 2. Central location error (pixels). Red fonts indicate the best performance while blue fonts indicate the second best.

	Struck	TLD	L1APG	CSK	Frag	KCF	DSST	RPT	FCNT	Ours
car4	8.7	12.8	77	19.1	119.1	9.9	2.7	8.1	4.7	3
carDark	1	27.5	1	3.2	37.7	6	2.7	4.2	4.3	1.1
crossing	2.8	24.3	63.4	9	57.7	2.2	2.8	1.7	5.2	1.5
david	42.8	5.1	14	17.7	93	8.1	7.9	6.6	5.9	5.1
david3	106.5	208	86	56.1	12.9	4.3	96.5	4.4	8.2	4.1
deer	5.3	30.9	24.2	5	111.8	21.2	4.4	4.2	7.7	3.7
jogging-1	7.9	5.2	89.5	135	27.6	88.3	102.7	120.3	6.2	4.9
jogging-2	136.2	7.3	145.8	164.7	33.6	144.5	154.6	161.3	15.3	6
liquor	91	37.6	212.9	160.6	102.2	5.3	46.1	49.9	30.1	4.4
singer1	14.5	8	53.4	14	77.1	12.8	3.6	10.8	5.5	6
skating1	82.9	145.5	158.7	7.8	138.4	7.7	6.5	8.4	14.4	8.2
walking2	11.2	44.6	5.1	17.9	57.3	29	59.2	16.5	5.9	3.6
avg	42.6	46.4	77.6	50.8	72.4	28.3	40.8	33	9.4	4.3

CSK [16], Frag [17], KCF [18], DSST [11], and RPT [19]. We get the tracking results of these compared methods from the benchmark [20]. To evaluate the effectiveness of feature fusion in target localization, we also compare our tracker with FCNT which using conv5-3 feature maps as default.

We use two criteria for quantitative evaluation: center location error (CLE) and overlapping rate (OR). OR is defined as $OR = \frac{|B_T \bigcap B_G|}{|B_T \bigcup B_G|}$, where B_T denotes the bounding box generated by trackers and B_G denotes the ground-truth bounding box. CLE is defined as the Euclidean distance between the center locations of B_T and B_G. The tracking results are summarized in Table 1 and Table 2. Overall, our tracker outperforms the other trackers signicantly in terms of both CLE and OR. In parparticular, the best performance of CLE can prove that feature fusion is more precise for target localization campared with FCNT. Figure 3 shows screen shots of tracking results from different trackers.

These sequences contain several challenging factors including background clutter, fast motion , scale and illumination variation , and occlusion, the results in the tables show that our tracker can handle those challenging factors effectively.

Background clutter. In the sequence Liquor, KCF and our tracker can follow the target till the end. Other trackers may lose the target for a while, frame #922 and #1609 show that the scale estimation of KCF is less accurate than out tracker. Our tracker achieves the best OR and CLE.

Faster motion. In the Deer sequence, due to fast motion, TLD, Frag, KCF and L1APG drift away at frame #12 and #31, our tracker achieves the best OR and CLE.

Scale and illumination variation. Frag and L1APG drift away in sequence Car4, David and Singer1. FCNT performs best in the sequence Car4, but in sequence Singer1, it suffers from less accurate scale estimation. Our tracker achieves a favorable performance compared with other trackers.

Occlusion. In Jogging-1 and Jogging-2, our tracker, FCNT and TLD can follow the target till the end. But their scale estimation is not so accurate as ours, so our tracker achieves the best OR and CLE in those sequences.

4. CONCLUSION

In this paper, we propose a novel CNN based tracking algorithm for robust visual tracking. Different from prior methods, we use a skip layer fusion to combine the features from different layers together to make target localization more accurate. We also hire a global detection network in case that the local detection network fails. Scale estimation correlation filter also contributes to the outstanding performance of our tracker. Applied to visual tracking ,we achieve favorable result against state-of-the-art methods in the challenging sequences.

5. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[5] Jialue Fan, Wei Xu, Ying Wu, and Yihong Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010.

[6] Naiyan Wang and Dit-Yan Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems*, 2013, pp. 809–817.

[7] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082.

[8] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han, "Online tracking by learning discriminative saliency map with convolutional neural network," *arXiv preprint arXiv:1502.06796*, 2015.

[9] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3119–3127.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[11] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[13] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 263–270.

[14] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 49–56.

[15] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1830–1837.

[16] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*. Springer, 2012, pp. 702–715.

[17] Amit Adam, Ehud Rivlin, and Ilan Shimshoni, "Robust fragments-based tracking using the integral histogram," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, 2006, vol. 1, pp. 798–805.

[18] 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.

[19] Yang Li, Jianke Zhu, and Steven C.H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[20] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.