A Modified SiamRPN for Visual Tracking

Wei Zhou, Yuxiang Liu, Haixia Xu, and Zhihai Hu

Abstract Siamese network based trackers have achieved state-of-the-art performance on multiple benchmarks. SiamRPN can predict the size of target thanks to RPN module. This paper proposes a modified SiamRPN based on IoU, under the framework of SiamRPN, Siamese feature extraction, and proposal generation for target, followed by the loss minimization. Aiming to the loss of both the classification branch and regression branch, we introduce IoU between GT box and the anchor into the regression loss function to form the joint optimization of IoU&smooth L_1 norm, which is useful to refine the tracking target box prediction. Then, we define IoU between GT box and the predicted box to weight positive samples. Weighted positive samples establish the connection between the classification branch and regression branch, which is helpful to eliminate the inconsistency in the optimal prediction of two branches. Experimental evaluations on the datasets OTB2013, OTB2015, demonstrate that compared with the state-of-the-art tracker such as SiamFC, SiamRPN and other algorithms, our proposed tracker achieves higher tracking accuracy and stronger robustness in most challenges of the tracking situation.

Keywords Target tracking · Siamese network · Intersection over union (IoU)

1 Introduction

In recent years, deep learning [1] has been leading the progress of visual tasks, such as target tracking [2], image segmentation [3] and target detection [4] and others. We focus on the target tracking, and also pay close attention to other tasks for they promote each other.

Trackers based on Siamese network have attracted many researchers thanks to their balance of speed and accuracy. Tao et al. [5] propose Siamese instance search for tracking (SINT), which adopts Siamese network structure, matching candidate image patches with multi-scale and the target patch. Then, Bertinetto et al. [6] design

W. Zhou · Y. Liu (✉) · H. Xu · Z. Hu
Xiangtan University, Xiangtan 411105, China

J. Yao et al. (eds.), *The International Conference on Image, Vision and Intelligent Systems (ICIVIS 2021)*, Lecture Notes in Electrical Engineering 813,
https://doi.org/10.1007/978-981-16-6963-7_70

the full convolution Siamese network, named SiamFC, which measures the similarity between the search image features and the target image feature through correlation convolution, and formulate the target tracking into the problem of the image matching. SiamRPN [7] introduces region proposal network (RPN) into the Siamese network, and utilizes the anchor mechanism of the object detect task [8] to predict the size of target. Therefore, a boundary box regression branch and a classification branch are added to SiamFC to discriminate the target and bound the target candidate region. Dsiam [9] explores a dynamic Siamese network to learn object appearance changes and background suppression online, and trains them with continuous video frames. DasiamRPN [10] uses the detection dataset to expand the positive samples and the difficult negative samples, and designs the interference perception module to distinguish the real target from the disturbance, which improves the generalization of the tracker.

SiamRPN implements the size prediction of the target by introducing RPN module, but several aspects are to be modified.

Firstly, the regression branch in SiamRPN is optimized by L_1 norm loss, so the prediction of bounding box is not accurate [11, 12].

Secondly, SiamRPN filters positive and negative samples through the Intersection over Union (IoU) ratio between anchor and the Ground Truth (GT) bounding box, which leads to low discrimination among positive samples.

Finally, the classification branch is separate from the regression branch in the introduced RPN module, which may not lock the same candidate target patch in the optimal prediction of the two branches.

In this paper, we propose a modified SiamRPN based on IoU. Under the framework of SiamRPN, we introduce IoU between GT box and anchors into the loss function to refine the regression prediction box, and define IoU between GT box and predicted box to weight positive sample for distinguishing each other, and positive samples based on IoU establish the connection between the classification branch and regression branch. Tracking experiments are carried out on OTB2013 [12], OTB2015 [13] test datasets to verify the feasibility and effectiveness of the proposed tracker.

The remainder of this paper is organized as follows. Section 2 discusses the principle of Siamese network. A modified SiamRPN for visual tracking is proposed in Sect. 3. In Sect. 4 experiments and discussion are given. The final section presents conclusion as well as future work.

2 Siamese Network

The classic Siamese network used in the tracking task is shown in Fig. 1. It formulates the problem of target tracking into the matching one between images.

Siamese networks apply an identical transformation φ to both exemplar image z and candidate image x, and measure the similarity between their representations by cross-correlation Layer as follows.

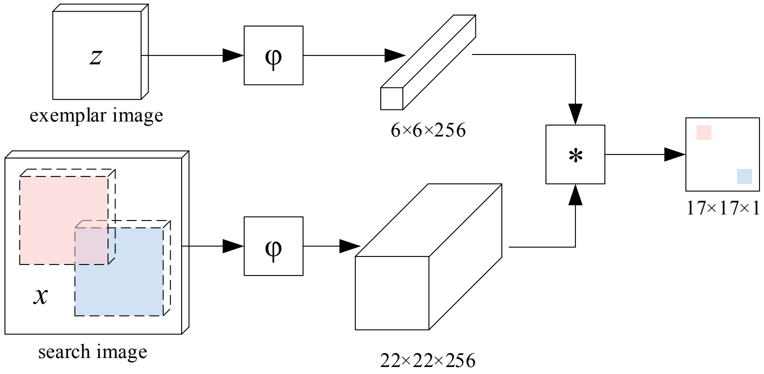

Fig. 1 Siamese network

$$f(z, x) = \varphi(z) * \varphi(x) + b \tag{1}$$

where b is a bias at every location of score map.

The similarity measure function $f(z, x)$ is learned to evaluate the similarity between the exemplar features and the candidate features, so as to obtain the similarity response score map that shows s a high score if the two images depict the same object and a low score otherwise.

3 The Proposed Method

In this section, we propose the modified Siamese-RPN based on IoU, illustrated in Fig. 2. Under the framework of SiamRPN, we introduce IoU between GT box and

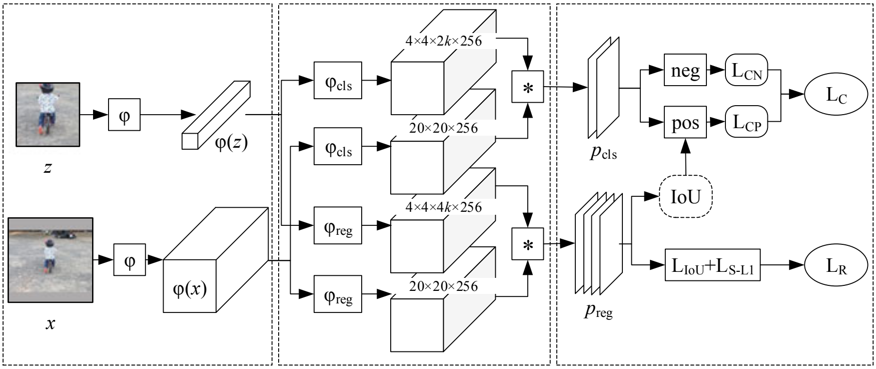

Fig. 2 A modified Siamese-RPN framework

anchors into the loss function to refine the regression prediction box, and define IoU between GT box and predicted box to weight positive sample for distinguishing each other, and positive samples based on IoU establish the connection between the classification branch and regression branch.

We divide it into four parts a Siamese feature extraction module, region proposal module, bounding box regression, and foreground–background classification.

3.1 Siamese Feature Extraction Module

Siamese feature extraction module is to map images into feature representation domain. As is shown on the left block of Fig. 2, it consists of two branches, one is for the feature extraction of exemplar image, which is from the historical frame, we denote input z, output $\varphi(z)$. The other is for search image which is from the current frame, we denote input z, output $\varphi(x)$.

They share the learnable network φ, which adopts a full convolution network layer of Alexnet [14]. That is, The input z with 127 × 127, got by center cropping and input x with 255 × 255 got, in the same manner, are fed into Siamese module for feature extraction.

3.2 Region Proposal Module

The region proposal module is to obtain proposal generation for tracking targets. As is shown in the middle block of Fig. 2, it has two Siamese convolution networks φ_{cls}, φ_{reg}, used for foreground & background classification branch and target bounding box regression branch, respectively, and each of which is matched with a supervision section.

In order to get the feature in the identical representation domain, the two Siamese convolution networks φ_{cls}, φ_{reg}, are applied to features $\varphi(z)$, $\varphi(x)$, respectively, and output $\varphi_{\text{cls}}[\varphi(z)]$, $\varphi_{\text{cls}}[\varphi(x)]$ for classification, and $\varphi_{\text{reg}}[\varphi(z)]$, $\varphi_{\text{reg}}[\varphi(x)]$ for regression.

Then we perform the cross-correlation on the classification branch and the regression branch as below

$$\begin{aligned} p_{\text{cls}} &= \varphi_{\text{cls}}[\varphi(z)] * \varphi_{\text{cls}}[\varphi(x)] \\ p_{\text{reg}} &= \varphi_{\text{reg}}[\varphi(z)] * \varphi_{\text{reg}}[\varphi(x)] \end{aligned} \tag{2}$$

Here, $\varphi_{\text{cls}}[\varphi(z)]$ and $\varphi_{\text{reg}}[\varphi(z)]$ server as convolution kernel, $\varphi_{\text{cls}}[\varphi(x)]$ and $\varphi_{\text{reg}}[\varphi(x)]$ server as input signal in the cross-correlation layer.

Anchor mechanism is introduced to the tracking task. If there are k anchors, classification prediction p_{cls} 2 k channels, and regression prediction p_{reg} output 4 k channels.

3.3 Loss Function

In this section, as is shown on the right block of Fig. 2, we introduce IoU into loss function, and re-formulate the loss function of the regression branch and classification branch, respectively, as the following subsections.

We apply the strategy from SiamRPN [7] to pick positive and negative training samples: In terms of IoU between anchors and Ground truth box of target, positive samples are defined as anchors which has IoU > 0.6. and negative samples are defined as anchors which have IoU < 0.3. We limit at most 16 positive samples and totally 64 samples from one training pair, and optimize the loss function of bounding box regress on the positive samples, and loss function of classification on total samples. We set 5 anchors with the same area and the aspect ratios [0.33, 0.5, 1, 2, 3].

Regression Loss. It is not so effective to use only L_1 norm loss for the optimizer of the bounding box regression in the SiamRPN.

According to the works [11, 15] survey, IoU loss is one of the most effective evaluation, and is more accurate than that of the Ln norm loss in the bounding box regression. However, IoU loss has the difficulties of the highly nonlinear, multi-degree of freedom and the multiple zero gradient regions [16], it is hard to optimize IoU loss. Meanwhile, parameter imbalance exists in RPN module [17]. It is further hard to optimize IoU loss of the RPN network. I guess it may be the main reason why SiamRPN doesn't directly use IoU loss.

Here, we develop the bounding box regression prediction loss based on the joint optimization of IoU loss and smooth L_1 norm loss.

In order to overcome the difficulty of IoU loss, we optimize only the IoU loss of the best positive sample, and optimize smooth L_1 loss on the other positive samples. It is noted that the best positive sample is defined as the anchor that has the max IoU.

At the same time, the best positive sample is located in the central region. IoU loss will play a less important role in training process if only being optimized on the best positive sample. We illustrate the joint optimization of IoU loss & smooth L_1 norm loss processing in Fig. 2.

To begin with input, exemplar image z is got by center cropping. The search image x is got by cropping at a new center, which is shifted with random pixels. Then inputs z, x are fed into Siamese module and RPN module to output the prediction. The loss of target bounding box regression based on IoU & smooth L_1 is given as

$$L_{\mathrm{R}} = L_{\mathrm{best}} + \sum_{i \in \mathrm{pos}} L_{\mathrm{S\text{-}L_1}}\left(p_{\mathrm{reg}}^{(i)}\right) \tag{3}$$

where pos is all of positive samples except the best positive sample. $L_{\mathrm{S-L_1}}$ is smooth $\mathrm{L_1}$ loss, which is computed as SiamRPN [7]. The loss defined as on the best positive sample L_{best} is formulated by

$$L_{\mathrm{best}} = 1 - I_{\mathrm{IoU}}\left(b_{\mathrm{reg}}^{(\mathrm{best})}, \mathrm{gt}_{\mathrm{reg}}\right) + R_{\mathrm{penalty}}\left(b_{\mathrm{reg}}^{(\mathrm{best})}, \mathrm{gt}_{\mathrm{reg}}\right) \tag{4}$$

where $\mathrm{gt}_{\mathrm{reg}} = \left\{\left(x_{\mathrm{gt}}, y_{\mathrm{gt}}, w_{\mathrm{gt}}, h_{\mathrm{gt}}\right)\right\}$ is GT target bounding box, $b_{\mathrm{reg}}^{(\mathrm{best})} = \{(x_{\mathrm{b}}, y_{\mathrm{b}}, w_{\mathrm{b}}, h_{\mathrm{b}})\}$ is the predicted target bounding box on the best positive sample. $I_{\mathrm{IoU}}\left(b_{\mathrm{reg}}^{(\mathrm{best})}, \mathrm{gt}_{\mathrm{reg}}\right)$ is the IoU between $\mathrm{gt}_{\mathrm{reg}}$ and $b_{\mathrm{reg}}^{(\mathrm{best})}$. Penalty term of IoU loss R_{penalty} describes a constraint on bounding box, and it is calculated as Ref. [18]

$$R_{\mathrm{penalty}}\left(b_{\mathrm{reg}}, \mathrm{gt}_{\mathrm{reg}}\right) = \frac{\rho^2\left(b_{\mathrm{reg}}, \mathrm{gt}_{\mathrm{reg}}\right)}{C^2} + \alpha\nu \tag{5}$$

where $\rho\left(b_{\mathrm{reg}}, \mathrm{gt}_{\mathrm{reg}}\right)$ is Euclidean distance between $\mathrm{gt}_{\mathrm{reg}}$ and $b_{\mathrm{reg}}^{(\mathrm{best})}$. The weight coefficient $\alpha = \frac{\nu}{\left(1 - I_{\mathrm{IoU}}\left(b_{\mathrm{reg}}, \mathrm{gt}_{\mathrm{reg}}\right)\right) + \nu}$. ν is used to measure the similarity of length–width ratio between the ground truth box and the predicted box, and computed by

$$\nu = \frac{4}{\pi^2}\left(\arctan\frac{w_{\mathrm{gt}}}{h_{\mathrm{gt}}} - \arctan\frac{w_{\mathrm{b}}}{h_{\mathrm{b}}}\right)^2 \tag{6}$$

From Eqs. (4) to (6), it can be seen that IoU loss keeps the target accuracy to the most extent in such aspects of intersection, length–width and center distance.

Classification Loss. SiamRPN picks positive and negative samples based on IoU between GT box and anchors. There is only one target in each image for the single target tracking task, so the positive samples are all from the same target. It is hard to determine which positive sample approaches more to the true target when their IoU is close to each other.

On the other hand, regression branch is separate from classification branch in SiamRPN, which may not lock the same candidate target patch in the optimal prediction of the two branches.

In this paper, we define weight coefficients for positive samples based on IoU between GT bounding box $\mathrm{gt}_{\mathrm{reg}}$ and the predicted bounding boxes $b_{\mathrm{reg}}^{(\mathrm{pos})}$, which are returned by regression branch.

The weight is used to distinguish sampled positive samples from each other. Consequently, these weighted positive samples bridge classification prediction and regression prediction. It is helpful to overcome the inconsistence by establishing the connection between classification prediction and regression prediction.

Then we formulate the classification loss on negative samples and weighted positive samples as below

$$L_{\mathrm{C}} = L_{\mathrm{CP}} + L_{\mathrm{CN}} \tag{7}$$

The classification loss on weighted positive samples is given by

$$L_{\mathrm{CP}} = \sum_{i \in \mathrm{pos}} L_{\mathrm{CE}}\left(\eta_{\mathrm{scale}} \cdot I_{\mathrm{IoU}}\left(b_{\mathrm{reg}}^{(i)}, \mathrm{gt}_{\mathrm{reg}}\right) \cdot p_{\mathrm{cls}}^{(i)}, \mathrm{gt}_{\mathrm{cls}}^{(i)}\right) \tag{8}$$

where $L_{\mathrm{CE}}(x, y)$ is cross-entropy loss function, $\mathrm{gt}_{\mathrm{cls}}^{(i)}$ $p_{\mathrm{cls}}^{(i)}$ are the ground truth and predicted classification logits of the ith positive sample, respectively. $I_{\mathrm{IoU}}\left(b_{\mathrm{reg}}^{(i)}, \mathrm{gt}_{\mathrm{reg}}\right)$ is weight coefficient for the ith positive sample.

All of positive samples weights is scaled by a scalar η_{scale} to reduce the stochastic volatility of regression prediction. Based on IoU and prediction, η_{scale} is defined as

$$\eta_{\mathrm{scale}} = \frac{\sum_{i \in \mathrm{pos}} p_{\mathrm{cls}}^{(i)}}{\sum_{i \in \mathrm{pos}} I_{\mathrm{IoU}}\left(b_{\mathrm{reg}}^{(i)}, \mathrm{gt}_{\mathrm{reg}}\right) p_{\mathrm{cls}}^{(i)}} \tag{9}$$

The classification loss on negative samples is given as

$$L_{\mathrm{CN}} = \sum_{i \in \mathrm{neg}} L_{\mathrm{CE}}\left(p_{\mathrm{cls}}^{(i)}, \mathrm{gt}_{\mathrm{cls}}^{(i)}\right) \tag{10}$$

where neg denotes negative samples.

Finally, the total loss function on two branches is given as

$$L_{\mathrm{SUM}} = L_{\mathrm{R}} + L_{\mathrm{C}} \tag{11}$$

4 Experiments

In this section, we evaluate our proposed algorithm by conduct experiments on benchmark datasets OTB2013 [12], OTB2015 [13]. All the tracking results ensure a fair comparison.

4.1 Parameter Settings and Implementation Details

Parameter settings. All of experiments run on Ubuntu 18.04, Python3.6.12 and Pytorch1.6.0 platform with an Intel Xeon Gold 5122 CPU and a GeForce RTX 2080Ti GPU, memory 16 GB.

These parameters of Siamese module and RPN module are obtained by optimizing loss function in Eq. (11) with Stochastic Gradient Descent (SGD). We perform 50 epochs with mini-batch 32, the learning rate decreased 10^{-2} to 10^{-6} at each epoch.

Implementation details. During offline training phase, we train our proposed Siamese-IoU through end-to-end on datasets GOT10K [19] and on YouTube-Bounding-Boxes [20]. During online tracking phase, there is no online adaptation since we formulate online tracker as one-shot detector.

4.2 Quantitative Analysis

Our proposed tracker is evaluated and compared with top other trackers SiamFC [6], SiamRPN [7], Staple [21], KCF [22], CSRDCF [23], STRCF [24]. Here, trackers SiamFC and SiamRPN are trained offline with the above parameter settings and implementation details, and tracked online with their default hyperparameters.

Evaluation criteria. (1) precision, report the ratio of successful frames which Euclidean distance between the center of the predicted bounding box and the center of the ground truth is less than the given threshold τ (τ is set to 20 pixels) to the total number of video frames. (2) success rate: report the ratio of the number of frames whose overlap score is greater than the given threshold (τ is set to 0.5) to the total number of video frames.

Result on OTB2013. OTB2013 datasets contain 50 video clips. The performance is evaluated in terms of success plot and precision plot. The tracking results are reported on the test sets of OTB22013 in Fig. 3. It can be seen that the tracker Ours achieves an average precision 88.1% and a success rate of 63.4%. Tracker Ours is superior to other trackers SiamRPN, SiamFC, Staple, KCF, CSRDCF STRCF.

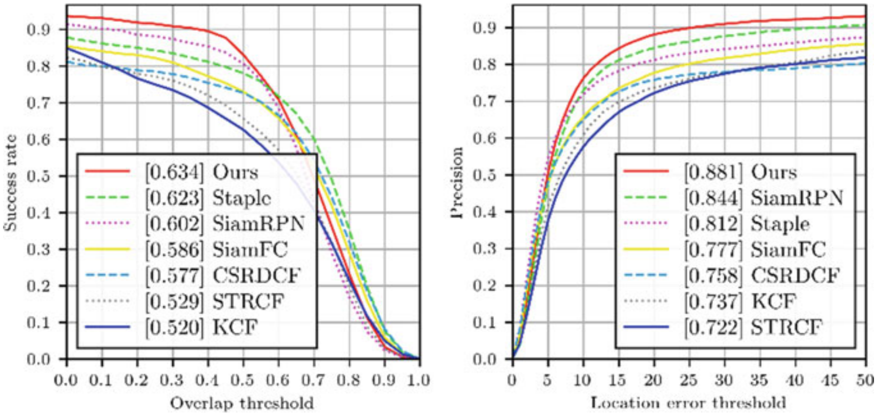

Fig. 3 Success plots and precision plots on OTB2013

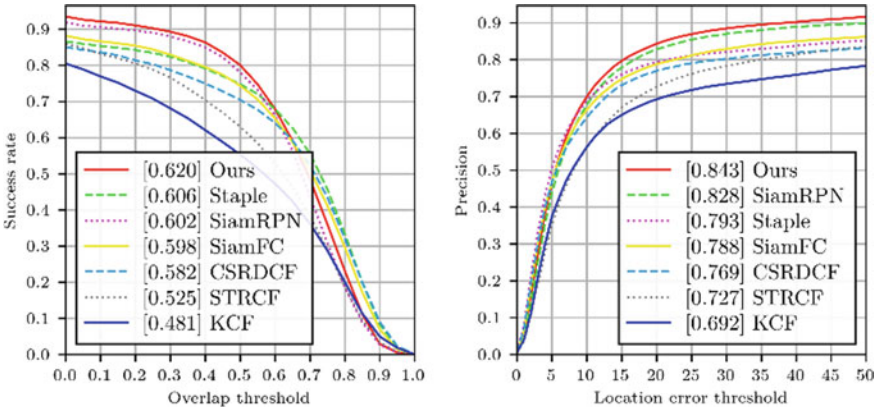

Fig. 4 Success plots and precision plots on OTB2015

Compared with top tracker SiamRPN, tracker ours increases by 3.2% and 3.7% in precision and success, respectively.

Result on OTB2015. OTB2015 datasets contain 100 video clips. The tracking results of the success plot and precision plot are illustrated in Fig. 4 on the test sets of OTB22015. It can be seen that the tracker ours achieves average precision 84.3% and success rate 62.0%. Tracker Ours is superior to other trackers SiamRPN, SiamFC, Staple, KCF, CSRDCF STRCF. Compared with top tracker SiamRPN, tracker Ours increase by 1.5% and 1.8% in precision and success, respectively.

To sum up, our proposed tracker (SiamIoU) outperforms significantly overSiamRPN, SiamFC and others in accuracy and EAO.

4.3 Qualitative Analysis

To intuitively evaluate and demonstrate Tracker Ours, we visualize the tracking comparison with SiamRPN, SiamFC on the following challenging clips from OTB2013, Lemming, Shaking, Singer2 and Ironman in Fig. 5. We give a brief qualitative analysis of the tracking visualization.

For the challenges of the Background Cluster (BC), Illumination Variation (IV), as can be seen in the sequence of Shaking, Singer and Lemming. The tracker Ours shows better robustness to IV than SiamRPN and SiamFC, for instance, the results of the frames that happen to flashlight on the clip Shaking. Our tracker bounds the target well thanks to the introduction of the IoU refine.

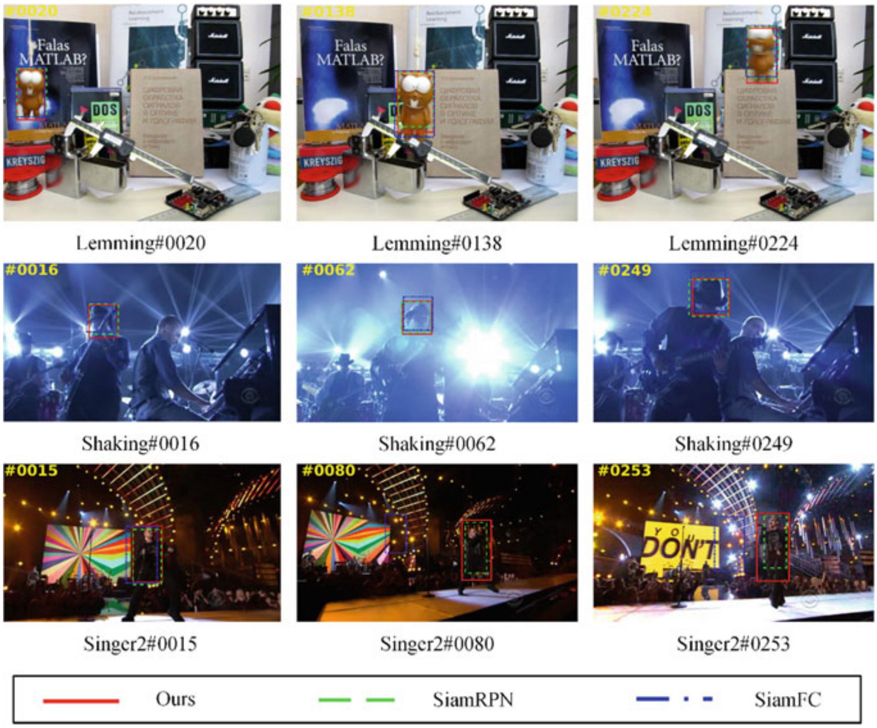

Fig. 5 Comparison of the tracking results of Ours with SiamRPN and SiamFC

5 Conclusion

In this paper, we propose a modified Siamese region proposal network based on the IoU, It is end-to-end offline trained on datasets GOT10K and YouTube Bounding-Boxes by applying box refinement procedure. In the inference phase, Our tracker is formulated as a local one-shot detector, and outperform SiamRPN and other trackers on datasets OTB2013, OTB2015.

Acknowledgements This work was supported by the Science and Technology Plan Project of Hunan Province (2016TP1020), open fund project of Hunan Provincial Key Laboratory of Intelligent Information Processing and Application for Hengyang normal university (IIPA20K04).

References

1. Zhang, R., Li, W., Mo, T., et al.: Review of deep learning. Inf. Control **47**(4), 385–397 (2018)
2. Hou, Z., Dai, B., Hu, D., et al.: Robust visual tracking via perceptive deep neural network. J. Electron. Inf. Technol. **38**(7), 1616–1623 (2016)

3. Li, D., Zhang, Z.: Improved U-Net segmentation algorithm for the retinal blood vessel images. Acta Opt. Sin. **40**(10), 101–110 (2020)
4. Guo, Z., Song, P., Zhang, Y., et al.: Aircraft detection method based on deep convolutional neural network for remote sensing images. J. Electron. Inf. Technol. **40**(11), 2684–2690 (2018)
5. Tao, R., Gavves, E., Smeulders, A.: Siamese instance search for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 1420–1429 (2016)
6. Bertinetto, L., Valmadre, J., Henriques, J., et al: Fully-convolutional Siamese networks for object tracking. In: European Conference on Computer Vision, Amsterdam, Netherlands, pp. 850–865 (2016)
7. Li, B., Yan, J., Wu, W., et al: High performance visual tracking with Siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, pp. 8971–8980 (2018)
8. Zhu, Z., Wang, Q., Li, B., et al.: Distractor-aware Siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, pp. 101–117 (2018)
9. Ren, S., He, K., Girshick, R., et al.: Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
10. Guo, Q., Feng, W., Zhou, C., et al.: Learning dynamic Siamese network for visual object tracking. In: Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, pp. 1763–1771 (2017)
11. Rezatofighi, H., Tsoi, N., Gwak, J.Y., et al.: General-ized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA, pp. 658–666 (2019)
12. Wu, Y., Lim, J., Yang, M.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, USA, pp. 2411–2418 (2013)
13. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
14. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
15. Yu, J., Jiang, Y., Wang, Z., et al.: Unitbox: an advanced object detection network. In: Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, Netherlands, pp. 516–520 (2016)
16. Tychsen-Smith, L., Petersson, L.: Improving obj-ect localization with fitness nms and bounded iou loss. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, pp. 6877–6885 (2018)
17. Li, B., Wu, W., Wang, Q., et al.: Siamrpn++: evolution of Siamese visual tracking with very deep net-works. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA, 2019, pp. 4282–4291 (2019)
18. Zheng, Z., Wang, P., Liu, W., et al.: Distance-IoU loss: faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, New York, USA, 2020, pp. 12993–13000 (2020)
19. Huang, L., Zhao, X., Huang, K.: Got-10k: a large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intell. (2019)
20. Real, E., Shlens, J., Mazzocchi, S., et al.: Youtube-boundingboxes: a large high-precision human-annotated data set for object detection in video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, pp. 5296–5305 (2017)
21. Bertinetto, L., Valmadre, J., Golodetz, S., et al.: Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp. 1401–1409 (2016)
22. Henriques, J., Caseiro, R., Martins, P., et al.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2014)

23. Lukezic , A., Vojir, T., ˇCehovinZajc, L., et al.: Discriminative correlation filter with channel and spatial reliability. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 2017, pp. 6309–6318 (2017)
24. Li, F., Tian, C., Zuo, W., et al: Learning spatial-temporal regularized correlation filters for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018, pp. 4904-4913 (2018)