Reimplementation of Siamese Region Proposal Network in visual tracking

University of Michigan
Junjiang Ye
junjiang@umich.edu

University of Michigan
Danny Chen
dannych@umich.edu

University of Michigan
Ziyu Guo
ziyuguo@umich.edu

Abstract

Visual object tracking is an essential foundation in various tasks of computer vision. And SiamRPN is a high-performance algorithm widely used for single object tracking problem. In this paper, we are trying to reimplement SiamRPN and evaluate this model. First, we will build the structure of SiamRPN, train and test this model in multiple datasets, including some new datasets which are not mentioned in original paper, such as GOT-10K. And then we will calculate the corresponding indicators of the trained model. In the end, we will compare the results with some other models to evaluate the performance and robustness of SiamRPN.

1. Introduction

Visual object tracking, a key branch of computer vision, has been used in various situations, such as automatic driving, industries, and video surveillance. SiamRPN [1], a high-performance algorithm used for single object tracking problem, performs better than many traditional algorithms in accuracy and robustness, such as Kernel Correlation Filter [2], ECO [3], etc.

In general, SiamRPN combines Siamese Network with Region Proposal Network, and it is divided to template branch and a detection branch, which are trained off-line with large-scale image pairs in an end-to-end manner. Then it formulates tracking as a local one-shot detection framework. SiamRPN is a novel algorithm, so we need to test and evaluate its performance in different situations. In this paper, we aim to reimplement SiamRPN. After training our model using ImageNetVID and YouTube-BB datasets, we then test our model on OTB2013 and OTB2015. What's more, we also test our model on GOT-10K dataset, which is a new one that has not been tested in original paper. Then, we compare the results with some other methods to verify its superiority.

The contributions can be summarized as three folds. 1). We reimplements SiamRPN. 2). We use multiple datasets to train the model. 3). It achieves leading performance in OTB2013, OTB2015 and a new dataset GOT-10K, which proves its advantages in accuracy and robustness.

2. Related Works

We provide a quick overview of three elements of our work because the SiameseRPN, which is phrased as a local one-shot detection job, is the core contribution of this study: trackers using RPN in detection, Siamese network structure, and one-shot learning

2.1 Trackers based on Siamese network structure

A Siamese network is made up of two branches that fuse together with a particular tensor to produce a single output after implicitly encoding the original patches to a different space. In the implicitly embedded space, it is typically used to compare the features of two branches, notably for contrastive tasks. Due to its balanced accuracy and speed, Siamese networks have recently attracted a lot of interest. Fully connected layers are used as the fusion tensor in GOTURN [4], which takes the Siamese network as the feature extractor. By using the last frame's anticipated bounding box as the only suggestion, it can be viewed as a regression approach. Siamese-FC [5] first presents the correlation layer as a fusion tensor and significantly increases accuracy, drawing inspiration from correlation - based approaches. When compared to GOTURN's single proposal regression, Siamese-FC and CFNet [6] is more resilient to fast-moving objects, which is the reason for its success. However, the fact that Siamese-FC and CFNet both require multi-scale testing and lack bounding box regression makes them less efficient. These real-time trackers' major issue is their inadequate accuracy and resilience when compared to cutting-edge correlation filter techniques.

2.2 RPN in detection

Faster R-CNN [7] is where the Region Proposal Network (RPN) was initially proposed. Previous proposal extraction techniques used before RPN take a lot of time. The proposal extraction is time-efficient while obtaining good performance thanks to the enumeration of multiple anchors [7] and sharing convolution features. Due to the supervision of both foreground-background categorization and bounding box regression, RPN is able to derive more precise recommendations. There are various RPN-based Faster R-CNN variations. To enhance the performance of tiny object detection, R-FCN [8] takes component position

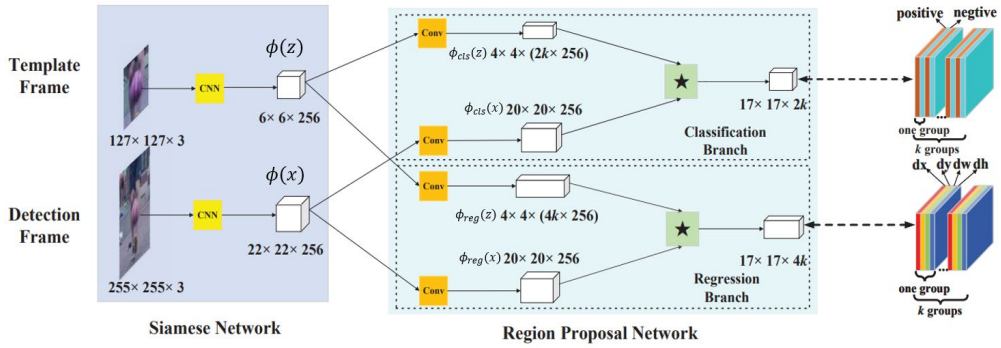

Figure 1: Main structure of Siamese-RPN: Siamese subnetwork for feature extraction is on the left side. The region proposal subnetwork, which has two branches and one for regression and the other for classification. In order to obtain the output of two branches, pair-wise correlation has been used. The right -side shows details of these two output feature maps. The output feature map for the classification branch comprises 2k channels that correspond to the foreground and background of k anchors. The resultant feature map in the regression branch has 4k channels, each of which corresponds to one of the four coordinates. Correlation operator is indicated in the figure by the symbol ⋆.

information into consideration while FPN [9] uses a feature pyramid network. The improved variants of RPN, including SSD [10] and YOLO9000 [11], are effective detectors as opposed to two stage detectors. Due to its quickness and excellent performance, RPN has numerous successful applications in detection; nevertheless, it hasn't been completely utilized.

2.3 One-shot learning

The one-shot learning subject in deep learning has received an increasing amount of interest in these years. The problem can be solved using two main approaches: meta-learning approaches and Bayesian statistics-based approaches. In [12], probabilistic models are used to represent object categories, and the inference stage use Bayesian estimation. The ability of learning to learn, or being conscious of one's own learning, is another goal of meta-learning. While [13] makes use of a neural network to estimate the gradient of the target network during the back-propagation, [14] develops a network that associates label with a small set of labelled support data and an unlabeled sample. Although these meta-learning-based algorithms have shown impressive results, relatively few of them have been applied to the tracking job. The tracking task, which predicts the parameters of a pupil network, is first addressed in Learnet [15]. However, Learnet's performance falls short of newer DCF-based approaches, such as CCOT, in several benchmarks.

3. Method

In this section, we describe the proposed Siamese RPN framework in detail. The suggested framework in Fig.2 is comprised of a region proposal subnetwork for proposal generation and a Siamese subnetwork for feature extraction. There are two branches in the RPN subnetwork, one of which is responsible for classifying foreground and background information and the other for anchor-proposal refinement. The proposed framework receives image patches containing the target objects, and the entire system is trained end-to-end.

3.1 Siamese feature extraction subnetwork

We use a completely convolution network without padding in the Siamese network. In order to meet the criteria for completely convolution with stride k, all paddings are eliminated. Let L_τ indicate the translation operator $(L_\tau x)[u] = x[u - r]$. Here, we employ a modified version of AlexNet [16], in which the conv2 and conv4 groups are unused [17].

There are two branches in the Siamese feature extraction subnetwork. The first one is referred to as the template branch and it gets the target patch from the first frame as input (denoted as z). The target patch from the current frame is the input (denoted as x). In CNN, two branches share parameters in order to embed two patches using the same transformation. We denote $\varphi(z)$ and $\varphi(x)$ as the output feature maps of the Siamese subnetwork.

3.2 Region proposal subnetwork

The pair-wise correlation and supervision sections make up the region proposal subnetwork. The supervision component has two branches, one for proposal regression and the other for foreground-background classification. The network must provide 2k channels for classification and 4k channels for regression. As a result, the pair-wise correlation section increases the channels of $\varphi(z)$ to $\varphi(z\_cls)$ and $\varphi(z\_reg)$, while splitting $\varphi(x)$ into two branches, $\varphi(x\_cls)$ and $\varphi(x\_reg)$, leaving the channels unchanged. The template feature maps $[\varphi(z)]$ is served as the correlation kernel. Softmax loss is employed to

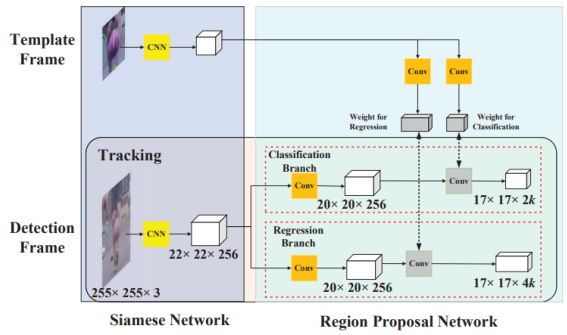

Figure 2: Tracking as one-shot detection: the template branch predicts the weights (in gray) for kernels of RPN on detection branch using the first frame. Then the template branch is pruned.

supervise the classification branch, while $2k$ channel vector is represented for negative and positive activation of each anchor at corresponding location on original map. Similarly, the 4k channel vector, which represents for dx, dy, dw, dh measuring the distance between anchor and corresponding ground truth and adopted cross-entropy and L1 loss with normalized coordinates for regression.

3.3 Training phase: End-to-end train Siamese RPN

Sample pairs are selected at random intervals from ILSVRC [18] and YouTube-BB [19] during the training phase. From two frames of the same video, the template and the detection patches are extracted. After the Siamese subnetwork was pretrained using ImageNet, we trained the Siamese-RPN end to end using stochastic gradient descent with data augmentations including affine transformation.

We select on fewer anchors for the tracking task than the detection task after realizing that the same item won't vary significantly in two adjacent frames. Thus, we just use one scale with several anchor ratios [0.33, 0.5, 1, 2, 3]. The methodology for choosing positive and negative training samples is also crucial. The criterion used in detection assignment is that we utilize IoU along with high (0.6) and low (0.6) thresholds as the measurement. The IoU above 0.6 will defined as positive and negative on the contrary. Additionally, we only allow up to 16 positive samples and 64 total samples from a single training pair.

3.4 One-shot detection

In order to predict the kernel of the local detection task, which is often the learning-to-learn process, we reinterpret the template branch in the Siamese subnetwork as training parameters. The pairwise bounding box supervision is the only supervision the meta learner requires throughout the training phase. In the inference phase, Siamese framework is pruned only leaving the detection branch except the initial frame. It will use the first frame to train the weights, which will be used as a kernel to in detection track. Then each frame after the first one will be convoluted by that kernel to get the response feature map, which is used to classify proposals. The target patch from the first frame is used to pre-calculate the category information and fixed during tracking period. As a result, it can be seem as one-shot detection.

We do anchor refinement, generate the top K proposals, and select them properly by the following two methods: First we discard the bounding boxes which are distant from the center by setting $g \times g$ subregion instead of $m \times n$. It successfully removes the outliers since nearby frames don't have massive motion. In the second proposition method, we reranked the proposals' scores using the cosine window and the scale change penalty to choose the best one. A cosine window is applied to suppress the big displacement after the outliers have been eliminated, and then a penalty is added to inhibit the large shift in size and ratio. Non maximum suppression (NMS) is performed afterwards to get the final tracking bounding box with linear interpolation size with target.

4. Experiments and Analysis

In this section, we set up several experiments to compare different models' performance [20]. We choose several datasets to Support our analysis. The OTB2015 dataset [21], also referred as Visual Tracker Benchmark, is a visual tracking dataset. It contains 100 commonly used video sequences for evaluating visual tracking, and each is annotated frame-by-frame with bounding boxes and 11 challenge attributes. The OTB2013 dataset [21] is the previous version of OTB 2015, which contains 51 sequences. We also choose GOT-10k [22], a large tracking database which offers a wide coverage of common moving objects in the wild. Specifically, GOT-10k introduces the one-shot protocol for tracker evaluation, where the training and test classes are zero-overlapped.

We have 1 original model and 1 modified model based on the original one. For the modified model, we change the number of channels of convolution layers in SiamRPN tracker. Specifically, the number of output channels of each convolution layer is amplified twice.

For the experiments, we use OTB2013 and OTB2015 to evaluate the performance of the original model. And use GOT-10k to evaluate the performance of the modified model. We remark that all the deep neural network models are trained. For the hyperparameters, we choose epoch=45, train-batch-size=32, valid-batch-size=8. And about the learning rate, we use waning learning rate from 0.04 to 0.00002.

There are two main experiments. First, we reimplement the SiamRPN model, calculate their success score and precision score, and draw the success plots to evaluate its performance under OTB2013 and OTB 2015 datasets. Second, we modify the SiamRPN tracker and retrain it with

same hyperparameters. Then we compare the performance between the modified model and other trending models.

4.1 Results of experiments

Experiment 1

Datasets	Success Score	Precision Score
OTB2013	0.631	0.842
OTB2015	0.621	0.825

Table 1. The evaluation of the SiamRPN model with OTB2013 and OTB2015 datasets.

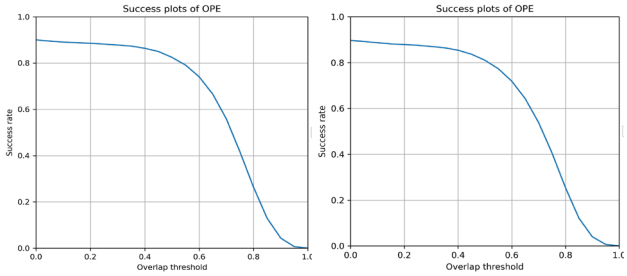

Figure 3. Success plot of the SiamRPN model on OTB2013 (left) and OTB2015 (right).

Experiment 2

Datasets	AO	$SR_{0.50}$	$SR_{0.75}$
GOT-10k	0.451	0.547	0.214

Table 2. The evaluation of the modified model with GOT-10k dataset.

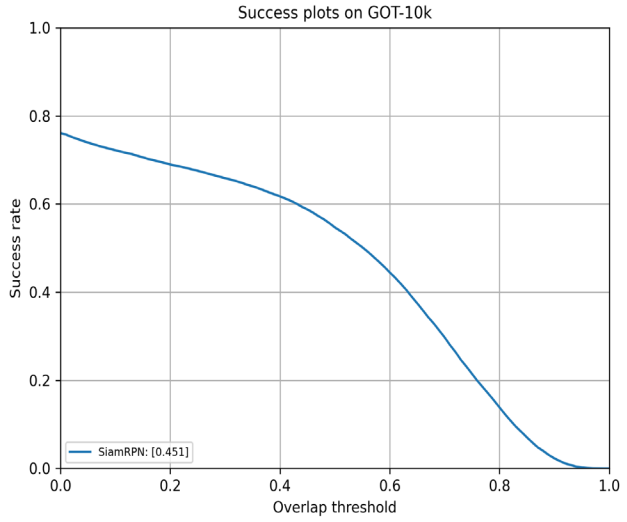

Figure 4. Success plot of the modified model on GOT-10k.

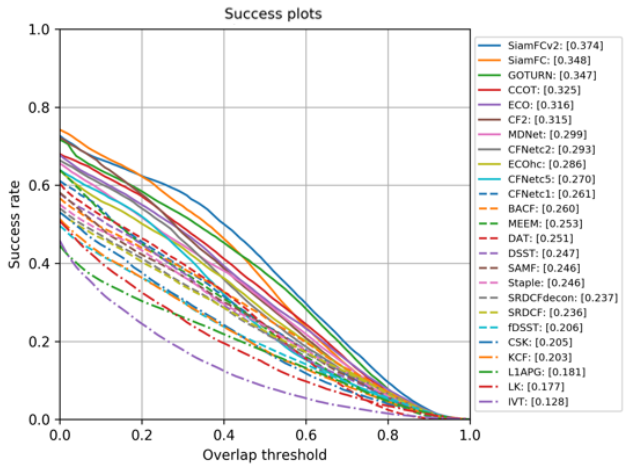

Figure 5. Success plots of some trending models on GOT-10k[1]

4.2 Analysis

In OTB2013 and OTB2015 test, we can find that SiamRPN performs well on these two datasets according to the success plots (Figure 3). The success scores are both above 0.6 and the precision scores are both above 0.8, which also match the results in the original paper [1].

The GOT-10k test is not implemented in the original paper [1]. In GOT-10k test, we can find that compared to other trending algorithms (Figure 5), the success plot of our model (Figure 4) performs better and can rank $1st$ among all the algorithms. And our average of overlap rates is higher than other models. Therefore, we can find that SiamRPN can still works well when using new datasets. And after we change the size of output channel, the performance of the model is still good enough.

Based on our results, we propose several future works.

5. Conclusion

In this paper, we have presented several methods on the object tracking task, including conventional and deep learning-based methods. Our method achieves better tracking precision comparing to the conventional method.

Based on our results, we propose several future works.

ResNet-based network Siamese trackers still have an accuracy gap compared with state-of-the-art algorithms and they cannot take advantage of features from deep networks, such as ResNet. Overcome spatial invariance restriction and train a ResNet-driven Siamese tracker to get a more efficient visual tracking model.

Increase Computational Power We need to have more computational power and larger datasets to train the general model. Due to the limited computational power and time, we only train our model at restricted batch-size and epoch. Result could be better if we have more powerful GPU to train the model.

[1] Plot Source: http://got-10k.aitestunion.com/index

References

[1] Li, Bo, et al. "High performance visual tracking with siamese region proposal network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[2] Li, Yang, and Jianke Zhu. "A scale adaptive kernel correlation filter tracker with feature integration." European conference on computer vision. Springer, Cham, 2014.

[3] Danelljan, Martin, et al. "Eco: Efficient convolution operators for tracking." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[4] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In European Conference on Computer Vision, pages 749–765, 2016.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional Siamese networks for object tracking. In European Conference on Computer Vision, pages 850–865, 2016.

[6] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. End-to-end representation learning for correlation filter-based tracking. The IEEE Conference on Com-puter Vision and Pattern Recognition, 2017.

[7] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In International Conference on Neural Information Processing Systems, pages 91–99, 2015.

[8] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 379–387. Curran Associates, Inc., 2016.

[9] T. Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. The IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In European Conference on Computer Vision, 2016.

[11] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. The IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[12] F. F. Li, R. Fergus, and P. Perona. One-Shot Learning of Object Categories. IEEE Computer Society, 2006.

[13] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. neural information processing systems, pages 3981–3989, 2016.

[14] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap. Meta-learning with memory-augmented neural networks. In International conference on machine learning, pages 1842–1850, 2016.

[15] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In Advances in Neural Information Processing Systems, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In International Conference on Neural Information Processing Systems, pages 1097–1105, 2012.

[17] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional Siamese networks for object tracking. In European Conference on Computer Vision, pages 850–865, 2016.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[19] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. Youtube-boundingboxes: A large high-precision human annotated data set for object detection in video. arXiv preprint arXiv:1702.00824, 2017.

[20] HonglinChu (2021) SiamTrackers [Source code]. https://github.com/HonglinChu/SiamTrackers/tree/master/SiamRPN/SiamRPN

[21] Y. Wu, J. Lim and M. -H. Yang, "Object Tracking Benchmark," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1834-1848, 1 Sept. 2015.

[22] Lianghua Huang et al. “GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.