# ShuffleNet_based Siamese Networks for Object Tracking

Mingxin Li[1,a], and Tianqi Yang[1,b]

[1]School of Information Science and Technology, Jinan University, Guangzhou 510632, China.

[a]2967664378@qq.com, [b]y_tq@163.com

## Abstract

**As one of the research hotspots of computer vision technology, object tracking plays an important role in many fields, such as intelligent monitoring and human-computer interaction. Researchers in different fields have proposed many different tracking algorithms. The SiamFC (Siamese Fully-Convolutional) has received lots of attention since it was raised. However, the accuracy and the discrimination ability of this algorithm is not enough. In this paper, we build a Siamese Networks based on ShuffleNet, named SF_Siam, for object tracking. The basic network of SF_Siam is ShuffleNet, which is composed of no-padding inside cropped residual units. In order to discuss the effect of no-padding inside cropped residual units, 3 residual units are proposed and discussed in this paper. Some empirical results are shown in the experiment part. Comparing with the state-of-the-art trackers, the proposed tracker could achieve comparable performance in multiple benchmarks.**

## Keywords

**Object Tracking; Siamese Network; Similarity Learning; ShuffleNet.**

## 1. Introduction

Object tracking is one of the most fundamental and challenging tasks in computer vision. Given the size and position of the target in the first frame, the task of object tracking is to predict the size and position of the target in all the following frames in a video sequence [1,2]. Target tracking technology plays an important role in missile guidance, intelligent monitoring systems, video retrieval, automatic driving, human-computer interaction and industrial robots. Although many tracking methods have made considerable progress in the past decade, it is still a very challenging task due to various negative scenes such as occlusion, fast motion, scale changes, and complex backgrounds [3].

The latest target tracking research methods are mainly based on two frameworks: discriminative correlation filters (DCF) and fully convolutional siamese networks. DCF uses the circular shifting of training samples and fast learning correlation filters in the Fourier frequency domain, which has good calculation performance and tracking accuracy. Therefore, DCF-based trackers has received widespread attention since MOSSE [4] first exploited it. However, most DCF-based trackers use offline pre-trained convolutional neural networks (CNN) for feature extraction, instead of using stochastic gradient descent (SGD) for online parameter fine-tuning. As a result, DCF based trackers benefit little from the end-to-end trainable networks. And the tracker based on the fully convolutional siamese networks uses the convolutional neural network to extract features and predicts the position of the target by comparing the similarity between the search area and the template area, showing great potential in high-performance visual tracking.

SiamFC [5] is a typical tracker based on fully convolutional siamese networks. It uses a fully convolutional network to extract the features of the target area and the search area, and then uses the cross-correlation operation to evaluate the search area to obtain the target location. It allows the SiamFC to obtain a greater improvement in speed and accuracy. However, even with a large amount of training data, SiamFC still has a performance gap to the best online tracker.

In this paper, we aim to improve the robustness and recognition accuracy of SiamFC. It is widely understood that, SiamFC uses the no-padding AlexNet [6] as the basic network structure. The depth of the AlexNet network is relatively shallow, and the features extracted by AlexNet is lack of discrimination. To overcome this restriction, we build a Siamese Networks based on ShuffleNet [7], named SF_Siam, for Object tracking. Different from standard ShuffleNet, our network is composed of no-padding inside cropped residual units. We tested our tracker on OTB2013 [8], OTB2015 [9], VOT2016 [10] and VOT2017 [11]. Results show that our tracker achieves satisfactory performance.

The rest of the paper is organized as follows. Section 2 introduces the most closely related works briefly. Section 3 describes the main part of the proposed approach. Section 4 carries out the experiment and presents the results while Section 5 draws a short conclusion.

## 2. Related works

### 2.1 Deep network

With the proposal of modern deep architecture AlexNet by Alex et al. [6] in 2012, research in network architectures is rapidly growing and many sophisticated deep architectures are proposed, such as VGGNet [12], GoogleNet [13], ResNet [14] and MobileNet [15]. These deep frameworks not only provide a deeper understanding of the design of neural networks, but also push forwards the state-of-the-arts of many computer vision tasks like object detection [16], image segmentation [17], and human pose estimation [18]. Similarly, the use of deep networks in object tracking has also achieved excellent tracking performance.

### 2.2 Siamese Network Based Trackers

Object tracking can be modeled as a similarity learning problem. By comparing the target image patch with the candidate patches in search region, we can track the object to the position that gets the highest similarity score. A significant advantage of this method is that it requires almost no training online and thence real-time tracking can be easily achieved.

The similarity learning of deep networks usually uses the Siamese network architectures. GOTURN [19] uses the Siamese network as feature extractor and uses fully connected layers as the fusion tensor. It can be seen as a regression method by using predicted bounding box in the last frame as the only one proposal. Re3 [20] employs a recurrent network to get better feature produced by the template branch. Inspired by correlation based methods, Siamese-FC [5] first introduces the correlation layer as fusion tensor and highly improves the accuracy. However, even if the SiamFC algorithm is trained with a large amount of data, its robustness and discrimination ability still have a certain gap compared with the state-of-the-art trackers.

There are a large number of follow-up work of SiamFC. CFNet [21] introduces correlation filters for low level CNNs features to speed up tracking without accuracy drop. And EAST [22] attempts to speed up the tracker by early stopping the feature extractor if low-level features are sufficient to track the target. SA-Siam [23] implemented a Siamese network with two branches, one branch is used for semantic feature extraction, and the other is used for appearance feature extraction. DaSi-amRPN [24] designed a new offline training sampling strategy. However, the above trackers use the no-padding AlexNet as the basic network structure. AlexNet cannot get deeper features, which have rich semantic information and is important for some special scenes such as motion blur and huge deformation. Therefore, this paper proposes SF_Siam to enhance the performance of SiamFC by improving the basic network.

## 3. Our Approach

Aiming at the limitations of the typical convolutional neural network structure in object tracking, we proposes Siamese Networks for Object Tracking based on ShuffleNet. However, simply training a Siamese tracker by directly using ShuffleNet does not obtain the expected performance improvement. The main reason is the intrinsic restrictions of the Siamese trackers. Therefore, we first give a brief analysis on the SiamFC.

### 3.1 Analysis on SiamFC

SiamFC formulates visual tracking as a problem of image similarity measurement. For the target image z and the candidate image x, the network learns a mapping function f(z, x). When the target image z and the candidate image x are the same target, f(z, x) gets a higher similarity score. Otherwise, f(z, x) gets a lower similarity score. The mapping function f(z, x) can be obtained by training with a large number of images. f(z, x) can be formula as:

$$f(z, x) = \varphi(x) * \varphi(z) + b1 \tag{1}$$

Where φ(.) denotes convolutional embedding function, b1 denotes a signal which takes value $b \in R$ in every location.

This simple matching function used in Siamese tracker has an intrinsic restriction for strict translation invariance, f(z, x[Δi]) = f(z, x)[Δi], where [Δi] is the translation shift sub window operator, which ensures the efficient training and inference. Li [25] found that padding in deep networks will destroy the strict translation invariance. In addition, the use of padding in the target tracking algorithm will also affect the prediction and positioning of the target. In the convolution process, adding padding to the input data is equivalent to adding noise around it. As the network gradually deepens and the convolutional layer increases, more and more noise is added around the original input data, which makes it easy for the network to generate a priori information: there is no target in the boundary area of the input data. Therefore, when the target moves to the boundary, it is prone to lose track targets.

### 3.2 ShuffleNet_based Siamese Networks for Object Tracking

The framework of the Siamese networks based on ShuffleNet (SF_Siam) is shown in Fig.1. The framework is divided into three parts, namely network inputs, feature extraction and similarity decision. The network inputs are the search image x and the target image z. And the image processing is similar to SiamFC. The feature extraction is mainly composed of an improved ShuffleNet. The similarity decision section is similar to the SiamFC, which uses a cross-correlation layer to measure the similarity between the target image and the search image. This paper focuses on the feature extraction.

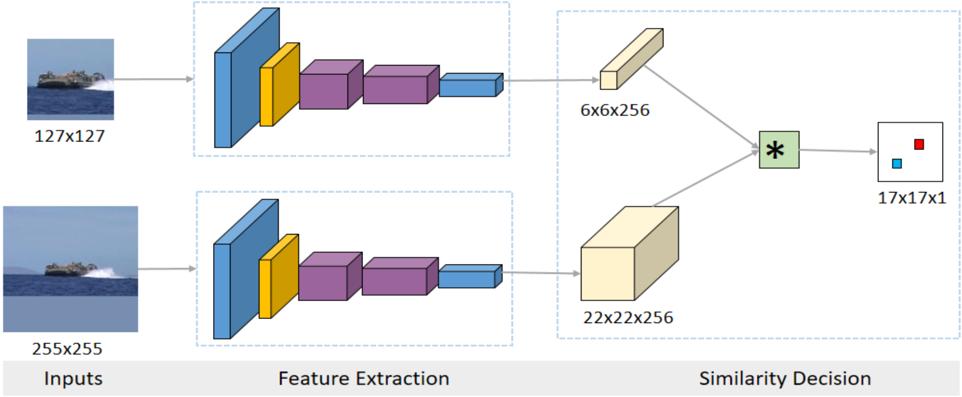

Fig. 1 The framework of the Siamese networks based on ShuffleNet (SF_Siam)

The core of ShuffleNet is the use of pointwise group convolution and channel shuffle, which reduces the amount of calculation while maintaining model accuracy. Therefore, this article will make improvements based on the ShuffleNet to make it more suitable for target tracking, which will be described in detail below.

As shown in Fig. 2(a) and (b), the basic unit of shuffleNet consists of four convolution layers and a jump connection layer, which is improved on a residual unit. The first convolutional layer on the right of the unit is a 1x1 group convolution, which groups the feature maps of the input layer, and then uses different convolution kernels to perform convolution operations on each group to reduce the amount of convolution calculations. The channel shuffle is added after the group convolution. It randomly mixes the output features of the previous layer to ensure that the input of the subsequent convolutional layer comes from different groups. And the last two convolutional layers are 3x3 depthwise convolution and 1x1 group convolution. At the end of the unit, a jump connection layer is used to connect the inputs and the outputs of the right convolutional layer. The function of the jump connection layer is to return the gradient in the neural network, so as to prevent the diffusion of the network gradient due to excessive network layers. When stride=1, the input of the unit is the same as the output of the right convolutional layer and can be added directly (as shown in Fig.2(b)). When stride=2, the size of the feature map output by the right convolutional layer is reduced, which is inconsistent with the input of the unit. To solve this problem, ShuffleNet uses a 3x3 avgpool operation with stride=2 on the inputs. Finally, connect the obtained feature map with the output of the convolutional layer. The unit of this structure enables ShuffleNet to greatly reduce the computational complexity of the model while maintaining accuracy.

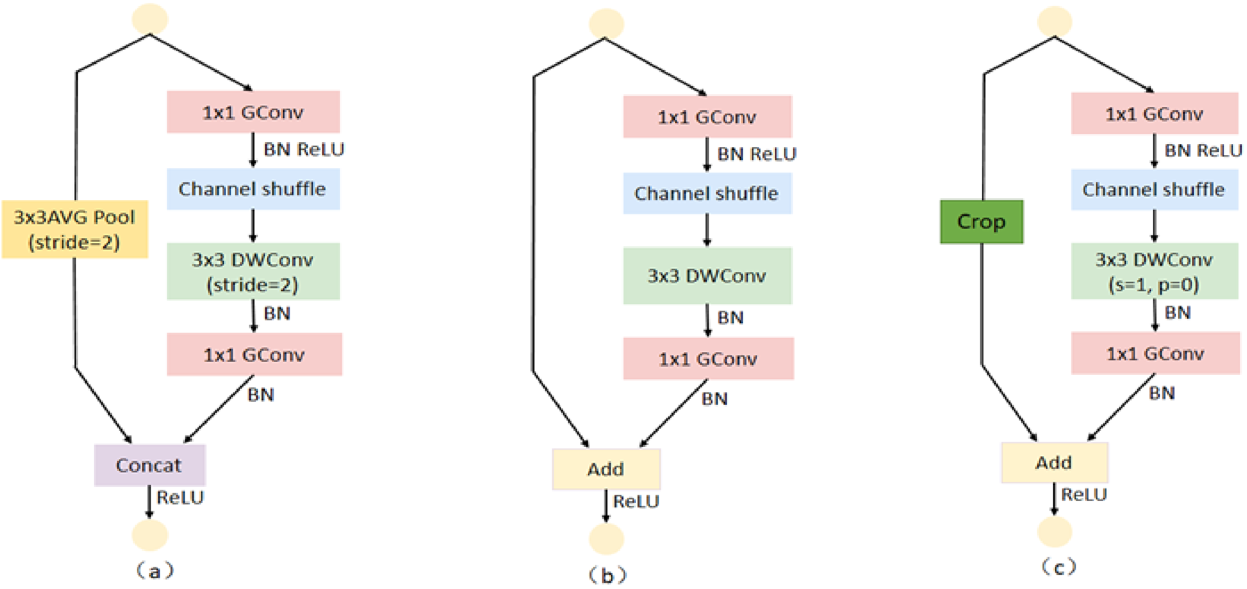

Fig. 2 ShuffleNet basic units

Since the padding of the deep network will destroy the strict translation invariance of the Siamese network, and it will also interfere with the target category prediction and positioning. We proposes an internal clipping residual unit based on the unit of ShuffleNet to make it better for object tracking.

As shown in Fig.2(c), the internal clipping residual unit is improved on the basis of Fig.2(b). Firstly, we change the 3x3 depthwise convolution with padding=1 to the 3x3 depthwise convolution with padding=0. Then crop the input feature to make it the same size as the output feature of the convolutional layer. Finally, add the input and the output of the convolutional layer. Finally, we use element-wise summation to merge inputs and convolutional layer outputs.

We use the internal clipping unit to build the network, and the dimensions of the parameters and activations are given in Table 1. It should be noted that in the ShuffleNet, we only modified the residual unit with stride=1, and the residual unit with stride=2 still uses the original unit like Fig.2(a). What’s more, other 3x3 convolutions on the ShuffleNet will no longer use padding.

Table 1. ShuffleNet architecture

| Layer | for search | for exemplar | KSize | Stride | Repeat | Output channels |
|---|---|---|---|---|---|---|
| Input | 255 x 255 | 127 x 127 | | | | 3 |
| Conv1 | 253 x 253 | 125 x 125 | 3 x 3 | 1 | 1 | 24 |
| Maxpool | 126 x 126 | 62 x 62 | 3 x 3 | 2 | | 24 |
| Stage2 | 62 x 62 | 30 x 30 | | 2 | 1 | 240 |
| | 58 x 58 | 26 x 26 | | 1 | 2 | |
| Stage3 | 28 x 28 | 12 x 12 | | 2 | 1 | 480 |
| | 22 x 22 | 6 x 6 | | 1 | 3 | |
| Conv2 | 22 x 22 | 6 x 6 | 1 x 1 | 1 | 1 | 256 |

### 3.3 Training the network

Similar to SiamFC, SF_Siam takes (z, x) as input. The convolutional network used to extract features is the modified ShuffleNet, and the features extracted are denoted by φ(.). The response map from the SF_Siam can be written as:

$$h(z,x)=corr(\varphi(z),\varphi(x)). \tag{2}$$

Where corr(.) is the correlation operation. Given the response map D ∈ R2 of the network, we suggest that an element u ∈ D is a positive sample if it is within radius R of the center:

$$y[u]=\begin{cases}+1 & \text{if } k\|u-c\|\leq R \\ -1 & \text{otherwise}\end{cases}. \tag{3}$$

Where k denotes the total stride of the network, c is the target center. For all training pairs, the corresponding labels are calculated by Eq.(3). We add a logistic loss layer to train the network at the end of the SF_Siam network:

$$l(y,v)=\log(1+\exp(-yv)). \tag{4}$$

Where y ∈ {+1, -1} represents the ground-truth label for each position u ∈ D in the score map. v denotes the real score of a single target-candidate pair returned by the model. We define the loss of a score map to be the mean of the individual losses:

$$L(y,v)=\frac{1}{|D|}\sum_{u\in D}l(y[u],v[u]). \tag{5}$$

The parameters of the network θ are obtained by applying Stochastic Gradient Descent (SGD) to the problem:

$$\arg\min_{\theta} E_{z,x,y}L(y,f(z,x;\theta)). \tag{6}$$

## 4. Experiments

### 4.1 Implementation Details

Our approach is trained offline on the GOT-10k [26] video dataset. Among a total of more than 10,000 sequences. It contains a majority of 563 object classes and 87 motion patterns, resulting in a scale of 1.5 million bounding boxes. The initial parameters of the network follow a Gaussian distribution, and are scaled according to the improved Xavier method [27]. Our network is trained with stochastic gradient descent (SGD). The gradients for each iteration are estimated with mini-batches of size 8, and the learning rate is decreased in log space from $10^{-2}$ to $10^{-5}$. We extract image pairs from GOT-10k by choosing frames with interval less than 100. And adopt exemplar images of 127×127 pixels and a search images of 255×255 pixels.

In order to solve the scale variation of the target, we search for the object over three scales $1.0375^{\{-1,0,1\}}$. And the score map was upsampled from 17×17 to 272 × 272 by using bicubic interpolation. Our method is implemented using TensorFlow on a GeForce GTX 1080 Ti.

### 4.2 Some analyses of the residual unit

In this section, we show some results and analyses of different residual unit of the ShuffleNet. In order to discuss the effectiveness of the internal clipping residual unit, we design 3 kinds of residual unit for ShuffleNet, one of them is internal clipping residual unit (as shown in Fig.2(c)), and the others are shown in Fig.3(a) and (b). They replace the internal cropping with 3x3 max-pool and 3x3 avg-pool respectively. We use these 3 kinds of residual units to build Siamese networks based on ShuffleNet and test them on OTB100 and OTB2013 respectively.

Fig.4 and Fig.5 shows the precision and success plots of SiamFC and the 3 kinds of SF_Siam architectures in OTB2013 and OTB100. In order to ensure the comparability of the experiment and better reflect the influence of different networks on the tracking results, SiamFC is also trained in GOT-10K. As shown in Fig.4 and Fig.5, compared with the other 2 network architectures, SF_Siam shows the best performance of both precision and success plots in all the two data sets. And compare with SiamFC, SF_Siam also has a significant improvement in both precision and success plots.

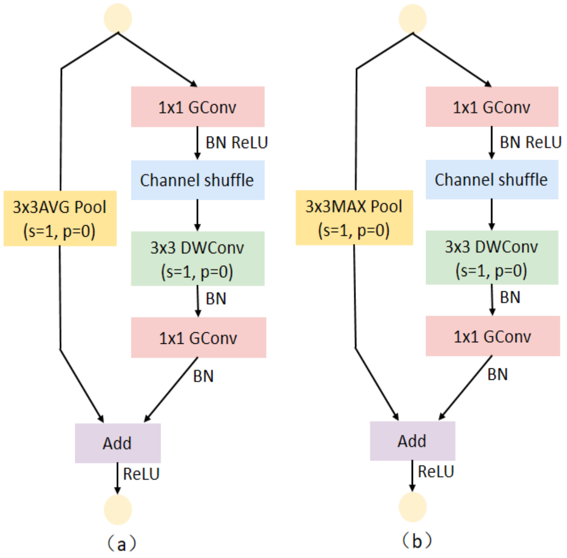

Fig. 3 Kinds of residual units

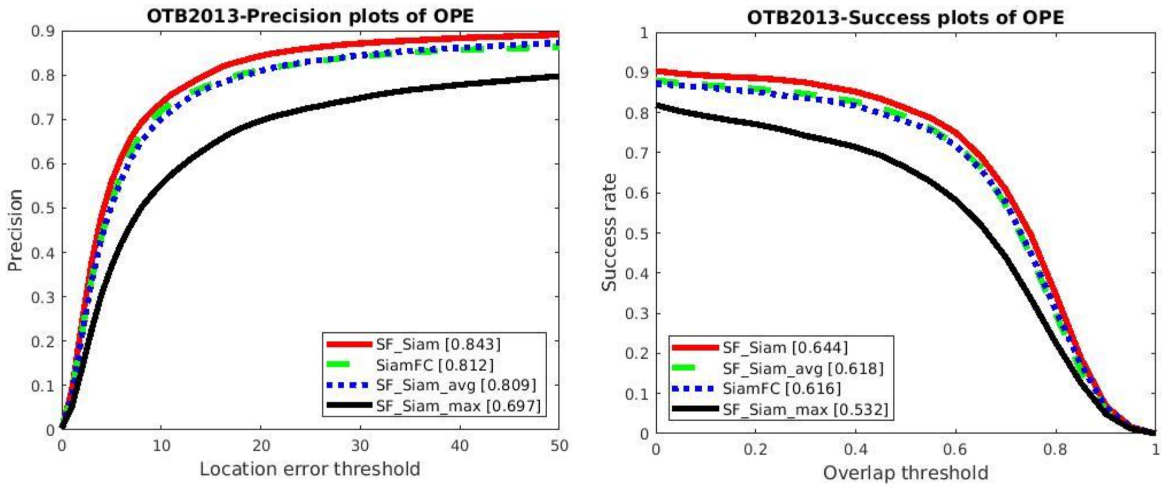

Fig.4 The precision and success plots of SiamFC and the 3 kinds of SF_Siam architectures in OTB2013.

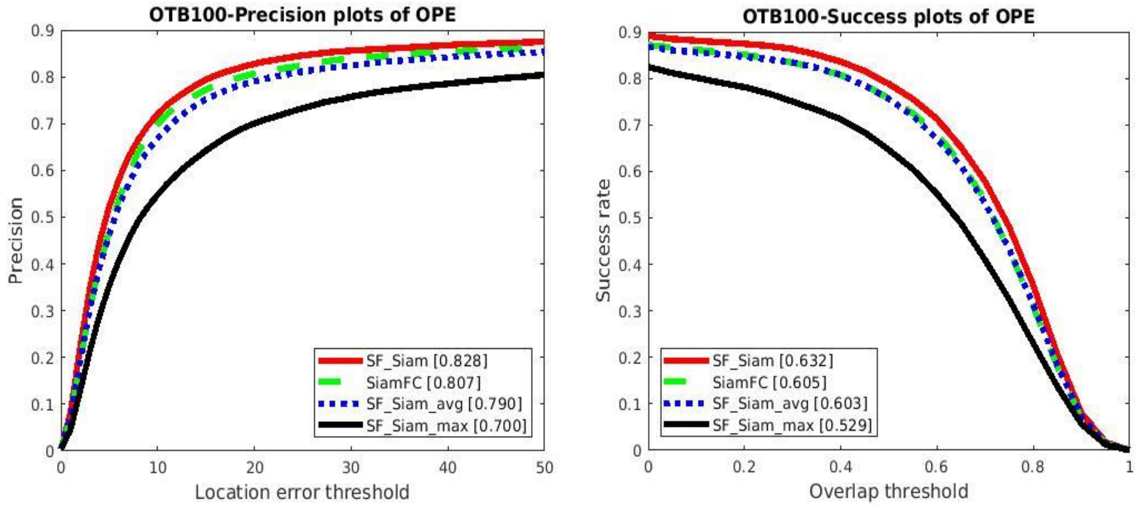

Fig.5 The precision and success plots of SiamFC and the 3 kinds of SF_Siam architectures in OTB100.

The results in Fig.4 and Fig.5 indicates that for the residual unit, although a small amount of features will be discarded by the clipping in the jump connection, it will not affect the final features extracted by the target. Because the main function of the jump connection is to return the gradient, the input of the jump connection are also used as the input of the right convolutional layer. The convolutional layer further extracts features from this input, and then passes them to the next structural unit through addition. Therefore, the main source of the output features of the residual unit is the convolutional layer on the right. But using pooling layer to reduce dimension will lose a lot of original information, which will affect the final feature extraction.

### 4.3 Result on OTB2015

OTB2015 contains 100 sequences collected from commonly used tracking sequences. And the two standard evaluation metrics on OTB2015 are success rate and precision. In this experiment, we compared SF_Siam with ECO-HC [28], SRDCF [29], Staple [30], DSST [31], SiamFC, SRDCFdecon [32], BACF [33], LCT [34] and LMCF [35] on OTB100. The precision plots and success plots of one path evaluation (OPE) are shown in Fig.6. The comparison shows that the algorithm proposed in this paper is 0.2% lower than ECO-HC in terms of Success scores and outperforms other state-of-the-art trackers.

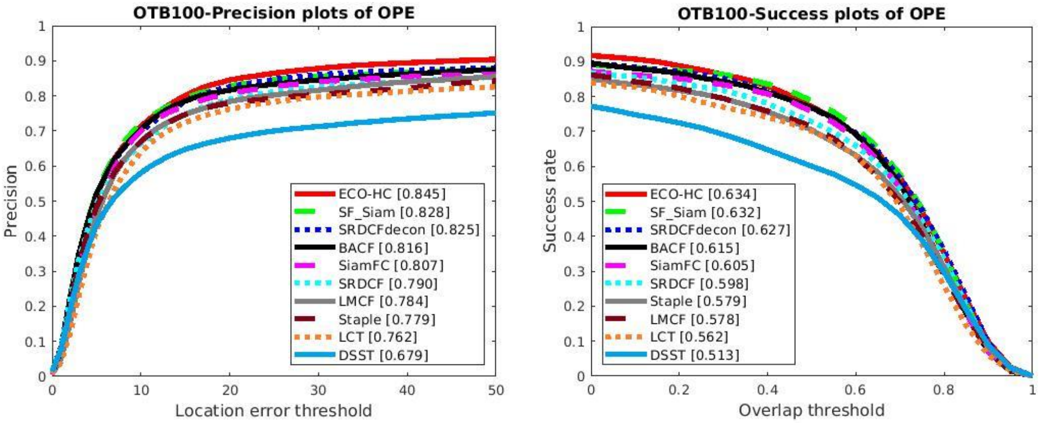

Fig. 6 Success plot and precision plot of OTB100

### 4.4 Result on VOT2016

The VOT2016 data set consists of 60 videos. It mainly evaluates the performance of a tracker based on accuracy and robustness. Accuracy is obtained by calculating the average overlap rate between the experimental results and the ground truths. And robustness is defined by the number of failures of the tracker. A failure is defined as when the overlap ratio between the predicted results and the ground truths is zero. Equivalent Filter Operations (EFO) is used to evaluate the speed of a tracker. And expected average overlap (EAO) is the most important indicator for VOT2016 to evaluate the accuracy of a tracker.

We compared the our method with other 9 trackers that have excellent evaluation results on the vot2016 data set, the results are shown in Table 2. SF_Siam outperforms other state-of-the-art trackers in terms of Overlap. In addition, all the performance of SF_Siam is better than its benchmark tracker SiamAN.

Table 2. Comparisons between SF_Siam and state-of-the-art trackers on VOT2016 benchmark.

| Tracker | SF_siam | CCCT | ColorKCF | DPT | GCF | NSAMF | SiamAN | SiamRN | STAPLEp | SSKCF |
|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.552 | 0.438 | 0.494 | 0.484 | 0.521 | 0.499 | 0.531 | 0.549 | 0.551 | 0.542 |
| EAO | 0.248 | 0.223 | 0.226 | 0.235 | 0.219 | 0.227 | 0.236 | 0.278 | 0.286 | 0.277 |
| Failures | 27.17 | 29.32 | 25.77 | 31.94 | 34.22 | 27.46 | 29.80 | 24.00 | 24.32 | 22.71 |
| FPS | 47.47 | 10.98 | 117.23 | 4.59 | 6.35 | 8.57 | 14.14 | 7.84 | 17.84 | 45.94 |

### 4.5 Result on VOT2017

Compared with VOT2016, VOT2017 replaces 10 challenging sequences with 10 difficult sequences. In addition, VOT2017 conducted a new real-time experiment in which the tracker needs to process a real-time video stream at least 25fps. It is challenging for almost all of the state-of-the-art trackers.

Table 3 shows SF_Siam along with several real-time trackers listed in the report of the VOT2017. Different trackers have different advantages, and SF_Siam can rank 1st according to AUC. In addition, SF_Siam outperforms SiamFC in terms of Accuracy, EAO and AUC.

Table 3. Comparisons between SF_Siam and state-of-the-art trackers on VOT2017 benchmark.

| Tracker | SF_siam | ASMS | ECOhc | KFebT | Mosse_ca | SiamDCF | SiamFC | ssckf | Staple | UCT |
|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.521 | 0.500 | 0.510 | 0.456 | 0.422 | 0.507 | 0.511 | 0.538 | 0.541 | 0.495 |
| EAO | 0.210 | 0.196 | 0.266 | 0.211 | 0.143 | 0.261 | 0.203 | 0.223 | 0.273 | 0.267 |
| AUC | 0.344 | 0.282 | 0.284 | 0.274 | 0.216 | 0.311 | 0.311 | 0.330 | 0.316 | 0.320 |

## 5. Conclusion

In this paper, we propose a new tracking method (SF_Siam) using ShuffleNet. In order to eliminate the influence of depth network padding on tracking algorithm which based on siamese network, a no-padding inside cropped residual unit is proposed in the architecture of the ShuffleNet. And to discuss the effect of the residual units, 3 kinds of residual units are proposed and discussed in this paper. Extensive experiments show that our proposed method achieves favorable performance against the state-of-the-art methods. In the future, we plan to continue exploring the effectiveness of deep networks in object tracking task.

## Acknowledgments

This work was supported by the Science and Technology Project of Guangdong (No.2017A010101 036), the Fundamental Research Funds for the Central Universities (No.21619413).

## References

[1] YANG H,SHAO L,ZHENG F,et al. Recent advances and trends in visual tracking:A review[J]. Neurocomputing, 2011,74(18):3823-3831. DOI:10.1016/j.neucom.2011.07.024.

[2] YILMA A,JAVED O,SHAH M. Object tracking:Asurvey[J]. ACM Computing Surveys, 2006, 38(4):1-DOI:10.1145/1177352.1177355.

[3] Li X, Liu Q, Fan N, et al. Hierarchical Spatial-aware Siamese Network for Thermal Infrared Object Tracking[J]. Knowledge Based Systems, 2017.

[4] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2544–2550.

[5] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H. Torr, Fully-convolutional siamese networks for object tracking, in: European conference on computer vision, Springer, 2016, pp. 850–865.

[6] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems, 2012: 1097-1105.

[7] Zhang X, Zhou X, Lin M, et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices[J]. 2017.

[8] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013: 2411-2418.

[9] Wu Y, Lim J, Yang M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9):1834-1848.

[10] Kristan M, Leonardis A, Matas J, et al. The Visual Object Tracking VOT2016 Challenge Results[C]. European conference on computer vision, 2016: 777-823.

[11] Kristan M, Alesˇ Leonardis, Matas J, et al. The Visual Object Tracking VOT2017 challenge results[C]// 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). IEEE, 2018.

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.2

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.2

[14] K. He, X. Zhang, S.Ren, and J.Sun. Deep residual learning for image recognition. In CVPR, 2016. 1, 2, 4, 6

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 2

[16] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. In CVPR, 2018. 2

[17] L.-C. Chen, Y. Zhu G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In ECCV, 2018. 2

[18] W. Tang, P. Yu, and Y. Wu. Deeply learned compositional models for human pose estimation. In ECCV, 2018. 2

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

[20] D. Gordon, A. Farhadi, and D. Fox. Re3: Real-time recurrent regression networks for object tracking. arXiv preprint arXiv:1705.06368, 2017.

[21] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. 2, 7

[22] C. Huang, S. Lucey, and D. Ramanan. Learning policies for adaptive tracking with deep feature cascades. In The IEEE International Conference on Computer Vision (ICCV), Oct2017. 2, 7

[23] He A, Luo C, Tian X, et al. A twofold siamese network for real-time object tracking[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4834-4843.

[24] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, W. Hu, Distractor-aware siamese networks for visual object tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 101–117 .

[25] Li B, Wu W, Wang Q, et al. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. arXiv preprint arXiv:1812.11703, 2018.

[26] Huang L, Zhao X, Huang K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[27] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: ICCV 2015. (2015)

[28] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. ECO: efficient convolution operators for tracking. In CVPR, 2017.

[29] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In International Conference on Computer Vision, 2015.

[30] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1401–1409, 2016.

[31] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference BMVC, 2014.

[32] Danelljan M, Häger, Gustav, Khan F S, et al. Adaptive Decontamination of the Training Set: A Unified Formulation for Discriminative Visual Tracking[J]. 2016.

[33] Kiani Galoogahi H, Fagg A, Lucey S. Learning background-aware correlation filters for visual tracking [C]// Proceedings of the IEEE international conference on computer vision. 2017: 1135-1143.

[34] Ma C, Yang X, Zhang C, et al. Long-term correlation tracking[C]// 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015.

[35] Wang M, Liu Y, Huang Z. Large Margin Object Tracking with Circulant Feature Maps[J]. 2017.