A lightweight convolutional network for infrared object detection and tracking

Yin Xu¹,² and Qiang Fan¹

¹Huazhong Institute of Electro-Optics, Wuhan National Laboratory for Optoelectronics, Wuhan 430223, China.

Abstract. As an important application of computer vision, visual tracking has been a fundamental topic. Compared with visible image, infrared image has the characteristic of low resolution, blurred contour and single color feature. Thus, it is still a challenge for infrared object tracking. Further, it is difficult to balance the real-time performance and accuracy. This paper proposed a method for target detection and tracking, with a deeper and lightweight MobileNet V2 structure as the backbone network. In the end, the tracker is tested on various datasets. Result shows that the tracker can get a balance between tracking accuracy and inference speed, which is crucial for deployment on mobile devices.

1. Introduction

Visual target tracking has a wide range of applications in autonomous vehicle navigation, video surveillance, military operations, environmental perception and aerospace remote sensing[1-4]. But when the target is in a complex background, it is difficult to distinguish the target from the background. In addition, various interference factors such as sudden irregular movement, illumination changes, changes in shape and posture during the target movement will cause tracking failure. Especially when partial occlusion or complete occlusion occurs during the tracking process, most existing tracking algorithms are difficult to track accurately.

At present, visual tracking algorithm can be roughly divided into two main branches. One branch is tracking based on correlation filter such as KCF[5] and ECO[6]. This branch uses ridge regression to train a classifier for target and background classification. Cyclic correlation is used to realize dense sampling and is transformed in Fourier domain to accelerate calculation. Yet, these trackers have limitations in real scenes employ. KCF has achieved excellent real-time performance, but tracking accuracy is significantly dropped when similar interfering objects appear. Compared with KCF tracking algorithm, ECO tracker improves the tracking accuracy, but it is difficult to meet the real-time requirements due to the complexity of calculation. Besides, another branch is based on siamese network, such as SiamFC[7], CFNet[8], SiamRPN[9]. SiamFC has two inputs, one is the template to be used as the benchmark, the other is the candidate sample to be selected. Target position is confirmed through the final calculation of the correlation response map. The algorithm successfully introduces the convolutional neural network into the target tracking field, achieving a good balance between frame-per-second (fps) and tracking accuracy. SiamRPN combines the Siamese network backbone network and the Regional Proposal Network (RPN), has good performance in terms of speed and accuracy. This method achieves a comprehensive surpass in fps and tracking accuracy than CF-based trackers. Nevertheless, by analyzing the network structure of SiamRPN, we can easily find
that the tracker still uses AlexNet or ResNet as the backbone for feature extraction. Compared with the lightweight network proposed in recent years, such as MobileNet v2[10], there is still much room for improvement in reducing the amount of computation.

Based on the above analysis, we use the lightweight network to replace the previous backbone network for feature extraction to reduce calculation time. At the same time, the template frame target coordinate supervision information is fully utilized. Meta-learning is used to train the model and initialize the model to a local optimal state. When faced with a new tracking task, the trained model can be automatically generated and adapted by learning the first frame of video supervision information.

2. Method and processing

2.1. Target detection

Human attention is the key point when dealing with a large amount of input information. It prioritizes the most valuable data by prioritizing limited resources to useful information. Corresponding to the visual attention behavior of human beings, the computer determines the importance of visual information by detecting the salience area when processing the input image[11]. Visual salience detection has been widely utilized in areas such as object detection, image compression, content-based image editing, etc., and is a very important fundamental topic in computational vision research. In the field of significant target detection research, the region-based salience detection method has become the mainstream method in the field due to its fast detection speed and high accuracy. The process of performing salience detection by such methods consists of two important steps: regional feature representation and contrast calculation. The effective representation of the features of the image region directly affects the quality of the salience map.

It can be told from the principle of the human visual attention mechanism that the target of interest often has a clear difference or a large contrast with the surrounding background. Therefore, this paper designs a frequency domain calculation method that uses the luminance feature estimation center and field contrast to highlight the real target, which can improve the detection accuracy and miss detection rate of weak targets. The steps for target detection are:

1. Normally, the image from the sensor are represent in format of RGB color space. However, to facilitate the salient analysis of the visible light image, the image from the RGB color channel should be converted to the Lab color channel for better emphasizing the color characteristics of the image objects.

2. Image noise can greatly undermine the accuracy of the target labelling. Gaussian filtering is used to linearly weight the whole image, which can effectively suppress the false alarm rate due to the existence of large amounts of random noise in the image.

3. Take the mean of value for all pixels of the image, denoted as $I_m$, and remove the mean value of each channel of the image. Then obtain the Euclidean distance from the result obtained by Gaussian filtering, and obtain a significant graph is:

$$I_s = \|I_m - I_{\text{gaussian}}(x, y)\|$$  \hspace{1cm} (1)

Where $\|,\|$ is the L2 norm.

4. Perform normalization on the significant graph $I_s$, and the target of the final detection.
2.2 Object tracking

The proposed tracker based on SiamRPN is illustrated in fig1. The siamese sub-network is composed of a template branch and a detection branch. The template branch takes the target block of the first frame of the video as input. The center of the target frame predicted in the previous frame of the detection branch is used as the reference point, and the clipping is performed as the input of the target block of the current frame. The siamese branches share CNN parameters. The left part is feature extraction backbone. The right part is sub-net RPN, consists of the classification branch and the regression branch. In the branch for classification, the output feature map has 2k channels, which correspond to the target and background (where k represents anchors, that is, the number of prediction boxes at each position). In the branch for regression, the output feature map has 4k channels, which correspond to the four coordinates of the k anchor points. Here, *represents cross-correlation, and the entire system is trained end-to-end. The channel number in the group is the same as the entire channel number. The correlation is evaluated for the classification branch and the regression branch respectively.

\[
A_{(w\times2k)}^{cls} = [\varphi(x)]_{cls} \ast [\varphi(z)]_{cls}
\]

\[
A_{(w\times4k)}^{reg} = [\varphi(x)]_{reg} \ast [\varphi(z)]_{reg}
\]  

The ROI generation process is roughly to generate k preset size boxes (Anchor) centered on each pixel of the image to be detected, check whether the target category is contained in the box through the classification branch. Then use the regression branch to determine the position of the Anchor position (x, y) and size (w, h) to adjust, according to the offset (dx, dy, dw, dh) to calculate the Bounding Box. Center point and shape of the anchor boxes are denoted by Ax, Ay, Aw, Ah and at meanwhile Tx, Ty, Tw, Th are corresponding values for ground truth boxes, the offset formula is as follows:

\[
dx = \frac{T_x - A_x}{A_x}
\]

\[
dy = \frac{T_y - A_y}{A_y}
\]

\[
dw = \ln \frac{T_w}{A_w}
\]

\[
dh = \ln \frac{T_h}{A_h}
\]

During training, the classification uses the cross-entropy loss function as the optimization objective function, and the regression branch uses the smooth L1 loss.

\[
loss = La + \mu Lb
\]
La is the loss function of classification branch and Lb is the position regression branch, $\mu$ is a hyper parameter. In the model prediction stage, the template branch output is pre-calculated and used as the core of the local detection, and is fixed during the entire tracking period. The detection branch uses the pre-calculated result to process the feature map of the current frame to achieve online detection. The detection branch is forwarded to obtain the classification and regression output, and the top k candidate regions are obtained according to the score of the candidate region. After generating the first k candidate regions, some candidate region selection strategies are used to suit the tracking task:

1. Abandon the prediction boxes generated by anchor points far away from the center point;
2. Use the cosine window and scale transformation penalty items to reorder the candidate regions;
3. Use non-maximum suppression (NMS) to select the final prediction frame.

Although the candidate regions are sorted by the above candidate region selection strategy, when multiple similar targets are contained in the background or occlusion occurs, the tracking is likely to fail. Therefore, a trajectory prediction strategy is added after the tracker.

2.3 Lightweight backbone

As we know, SiamRPN use such as Alexnet, VggNet and ResNet, which are suitable for target detection and recognition mission, as backbone networks. Yet, these network structures have a large amount of calculation and high redundancy, and are more suitable for high-precision floating-point calculations such as GPU. The mobile platform development is only suitable for fixed-point computing, and the computing power and power consumption are limited. Therefore, for the sake of reducing the computational expense, the lightweight MobileNet is selected as the backbone network for target feature extraction.

The fundamental component of MobileNet is the depth-level separable convolution. Depth-level separable convolution can actually be viewed as a decompose convolution operation, which is composed of two submodules: Depth-wise convolution and point-wise convolution. Depth-wise convolution differs from the standard convolution. For standard convolution, the convolution kernel operates on all of the input channels, while depth-wise convolution treats the input channels in a different way by conducting convolution using different kernels, each of which corresponds to a certain channel. Corresponding to an input channel, it is an operation in the depth direction. The point-wise convolution is the convolution using a 1x1 convolution kernel. Depth-wise convolution first uses depth-wise convolution to operate different input channels separately, and then uses point-wise convolution to combine whole outputs. In this way, the calculation expense and the parameter amount will be greatly reduced, and achieve the goal of keeping overall functionality similar as a standard convolution simultaneously.

3. Experiments

We employ PyTorch to train the deep learning model. It was trained on RTX 1080Ti Display cards. During the training, the size of input image pairs are set 112 pixels and 224 pixels respectively. While, a single 1080 Ti card was used to inference. The batch-size is set as 256. By using stochastic gradient descent (SGD) with an initial learning rate 0.001 to optimization. The training data is consists of the following dataset: ImageNet VID and UAV123. We set 40 training epochs.

The proposed method were compared with some typical trackers by some tougher video from LSOTB, the results are arranged in fig2 and fig3. It is kindly to find that the proposed tracker is closer to ground-truth in a complex background than other trackers. Also, the inferred result is more suitable for the real size of the target.
Finally, we compare the proposed tracker with SiamRPN using different backbone architectures. The results of ResNet-50 and AlexNet are listed in Table 1. It is easy to see that the proposed method can achieve comparable results with different backbones. The evaluation also suggests that the proposed method achieve a trade-off between accuracy and efficiency. Besides, the proposed tracker can achieve 190fps which superior than other trackers.

**Table 1.** Different tracker test results on LSOTB.

| tracker     | SiamRPN | SiamRPN | proposed |
|-------------|---------|---------|----------|
| Backbone    | ResNet-50 | AlexNet | MobileNet |
| accuracy    | 0.660   | 0.585   | 0.584    |
| precision   | 0.696   | 0.638   | 0.587    |
| fps         | 56      | 120     | 190      |

4. Conclusion

This paper designs a method for infrared target detection and tracking with lightweight fully convolutional siamese network. A simple and efficient salient detection method find the right object at first. Then, the MobileNet lightweight backbone network is used to replace other backbone networks, which obviously reduces the amount of calculation, to track the infrared target. Our method can track accuracy and ensure real-time requirements.
References

[1] Fiaz M, Mahmood A, Javed S, et al. Handcrafted and deep trackers: Recent visual object tracking approaches and trends[J]. ACM Computing Surveys (CSUR), 2019, 52(2): 1-44.

[2] Fan Q, Xu Y. A robust target recognition and tracking panoramic surveillance system based on deep learning[C]//AOPC 2019: AI in Optics and Photonics. International Society for Optics and Photonics, 2019, 11342: 113420A.

[3] Hu Y, Xiao M, Zhang K, et al. Aerial infrared target tracking in complex background based on combined tracking and detecting[J]. Mathematical Problems in Engineering, 2019, 2019.

[4] Fan Q, Lei B, Tan H. A robust anti-occlusion object tracking method[C]//Eleventh International Conference on Digital Image Processing (ICDIP 2019). International Society for Optics and Photonics, 2019, 11179: 111793K.

[5] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 37(3): 583-596.

[6] Danelljan M, Bhat G, Shahbaz Khan F, et al. Eco: Efficient convolution operators for tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 6638-6646.

[7] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking[C]//European conference on computer vision. Springer, Cham, 2016: 850-865.

[8] Valmadre J, Bertinetto L, Henriques J, et al. End-to-end representation learning for correlation filter based tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2805-2813.

[9] Li B, Yan J, Wu W, et al. High performance visual tracking with siamese region proposal network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8971-8980.

[10] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[11] Fan Q, Sun G, Zou E. Tracking of small infrared targets based on convolutional neural network[C]//Journal of Physics: Conference Series. IOP Publishing, 2020, 1507(9): 092001.