IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 9, 2020, accepted August 27, 2020, date of publication September 7, 2020, date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022351

FPSiamRPN: Feature Pyramid Siamese Network With Region Proposal Network for Target Tracking

YUNBO RAO[1], (Member, IEEE), YIMING CHENG[1], JUNMIN XUE[1], JIANSU PU[1], QIUJIE WANG[2], RIZE JIN[3], AND QIFEI WANG[4], (Member, IEEE)

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]School of Computer Science, Guangdong University of Technology, Guangzhou 510006, China
[3]School of Computer Science and Technology, Tianjin Polytechnic University, Tianjin 300384, China
[4]Department of Electrical Engineering and Computer Sciences (EECS), University of California at Berkeley, Berkeley, CA 94720, USA

Corresponding author: Yunbo Rao (raoyb@uestc.edu.cn)

This work was supported in part by the Science and Technology Service Industry Demonstration Project of Sichuan under Grant 2019GFW126; in part by the Science and Technology Project of Sichuan under Grant 2019YFG0504; in part by the National Natural Science Foundation of China (NSFC) under Grant 61872066, Grant U19A2078, and Grant 61806142; and in part by the Sichuan Science and Technology Innovation Project under Grant 2019006.

ABSTRACT Target tracking based on Siamese network has reached the state-of-the-art performance. However is still limited in semantic feature extraction. In this paper, we propose a novel method to distinguish positive and negative samples. Taking deep neural network as the backbone, we fuse the feature maps from different layers and feed it to RPN (Region Proposal Network). In addition, we use a loss term for loss function to achieve self-adjusting and learn more discriminative embedding features of target objects with similar semantics. In the tracking stage, one-shot detection is used as the reference, fix the first frame as the weight of tracking to track the subsequent frames. Our method has achieved outstanding performance on several benchmark data set, such as: OTB2015, VOT2016, VOT2018, and VOT2019 *et al.*

INDEX TERMS Target tracking, siamese network, feature pyramid, region proposal network.

I. INTRODUCTION

Visual target tracking has received more and more attention in the past decades and has been a very active research field. It has been widely used in two-way fields such as visual monitoring [1], human-computer interaction [2], pedestrian tracking [43], and augmented reality [3]. Despite recent advances, it is still recognized as a challenging task due to a variety of factors, including changes in light, occlusion, and background clutter.

In recent years, most of the visual tracking algorithms are related to deep learning [4], [5]. Compared to the correlation filtering methods [6]–[9], the deep learning methods are more popular. Especially in recent years, the single target tracking method based on the Siamese network [10]–[15] has attracted extensive attention in society. In the initial stage of offline, siamFC tracker [11] adopts the full convolution network structure to train the deep conv network, aiming to solve the more common similarity learning problem, and then conduct online evaluation during the tracking process.

In order to ensure the tracking efficiency, the Siamese similarity function of offline learning is usually fixed in the running time [10], [11]. CFNet tracker [13] and DSiam tracker [12] update the tracking model by running average templates and fast conversion modules respectively. The SiamRPN tracker [14] introduces the area recommendation network after the Siamese network and performs joint classification and regression for tracking. The DaSiamRPN tracker [15] further introduces an interference-sensing module and improves the recognition capability of the model. SiamRPN++ tracker [16] eliminates damage from translation invariance and breaks the limitation of space invariance when using deep networks. These Siamese trackers draw the visual object tracking problem as learning general similarity graph through the relationship between the feature representation learned in the target template and the search area. In SiamMask [17], the tracking of objects is actually the block of the object mask, so the object mask is extracted first, and

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

then the object is tracked according to the object's mask. SiamDW [18] are of different kinds of backbone were studied in detail, and points out the padding is how to affect the precision in the process of training, target tracking.ATOM's work [19] is divided into two tasks, classification task and assessment tasks. The classification task is to separate the foreground and the background image, get a rough idea of target location.Assessment task is through the bounding box,which can be used to predict the state of the target. By decomposing the visual tracking task into two sub problems: the classification of the pixel category and the regression of the object bounding box at that pixel, SiamCAR [44] proposed a novel full convolution twin network to solve the end-to-end visual tracking problem in a per-pixel way. Due to the parameter complexity caused by the introduction of RPN, SiamBAN [40] avoids many super-parameters and more flexible. Re-identification [41] article adopts the combination of identification loss and triplet loss, but rather than simply adding the weight loss coefficient before two losses as the total loss, it proposes its own dynamic loss training. Zhong *et al.* [42] proposed a hierarchical tracker, which is based on the combination of coarse-level data-driven search and fine-level coarse-to-fine verification to learn movement and tracking. At a rough level, the data-driven motion model learned from deep loop reinforcement learning provides a rough location for their tracker.

Although these Siamese trackers have gained excellent tracking performance, especially in terms of balanced precision and speed. But even well-performing Siamese tracker, such as SiamRPN [14],whose tracking accuracy of distractors still has significant difference on similar targets. However, these trackers operate on the cross-correlation between the feature maps generated on the two branches of the network. The proposed methods ignore the influence of the feature maps generated in the middle of the network on different categories of tracking objects. Under the inspiration triggered by this observation, we analyze the existing Siamese trackers and find out that the core reason is that the convolutional layer of different levels represents the target of different aspects, the deep feature map contains more semantic features and can be used as a similar category detector. The lower level contains more discriminant information and can better separate the target from the background.

In order to solve this problem and obtain a more generalized Siamese tracker, a feacture pyramid Siamese network is proposed and it is experimentally proven that feature maps of different depths have different representation features. Experimental results are shown in Figure 1. The deeper convolutional layer captures more semantic features of objects, while the lower layer provides more detailed external features to better distinguish objects from backgrounds. The proper fusion of these different features helps to separate the target from the interference term.

In addition, few of these methods can clearly put forward very effective solutions to distinguish between targets and interferences. Based on the problem, this paper proposes a loss function to increase the differentiating effect between targets and distractors, and experiments show that our method has certain effects on differentiating distractors.

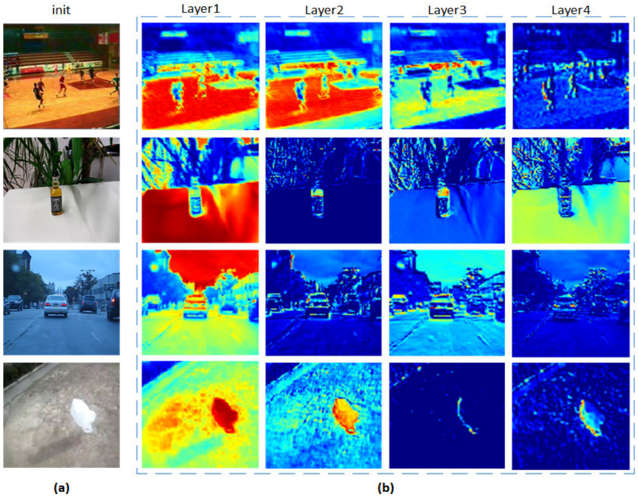

FIGURE 1. Heat map from our backbone network: (a) is the original image selected from OTB and VOT datasets, and (b) is the heat map output through our network, each column comes from a certain convolution layer in different blocks.

Our main contributions are summarized as follows:

1. The feature pyramid fusion method is proposed, which combined with the deep network backbone to retain more tracking target features.

2. Depth-wise cross correlation is adopted to solve the asymmetric problem, and loss items embedded in discriminating instances are introduced into the loss function for distinguishing objects with the same semantic class or similar appearance.

3.The proposed trained tracking network is tested on OTB2015, VOT2016, and VOT2018. The expected average overlap on VOT2016 improved by 2.9% compared to SiamRPN. The precision on OTB2015 improved by 5.3% compared to DaSiamRPN.

II. RELATED WORKS

We mainly introduced the recent trackers, especially those based on Siamese networks and briefly reviewed three aspects related to our works: deep network trackers based on Siamese networks, RPN detection and pyramid feature extraction, and one-time learning.

A. DEEP NETWORK ANALYSIS

Recently, Siamese networks have attracted great attention in the field of visual tracking because of their balanced accuracy and speed [11]–[13], [20], [21]. GOTURN [21] used Siamese network as feature extractor and fully connected layer as fusion tensor. By using the prediction bounding box in the last frame as the only proposal, it is a regression method. Re3 [20] used circular networks to get better functionality from template branching. Inspired by relevant methods, SiamFC [11] introduced relevant layers as tension tensors and greatly improved the accuracy. CFNet [13] added a correlation filter to the template branch, making the Siamese network shallower but more efficient. However, both SiamFC and CFNet use shallow layers network.

Since the shallow layer network cannot fully obtain the feature information of the object, it is hard for previous Siamese network tracker [10], [11] to achieve good performance, such as single and one-sided feature extraction. But some research have shown that training Siamese trackers simply by using deeper networks. Such as ResNet does not contribute to improvements of performance. SiamRPN++ [16] points out that there are two inherent limitations when utilize deep network for tracking training:

1) The contraction part and feature extractor used in Siamese tracker have inherent limitations on strict translation invariance. as (1) shows:

$$f(z, x) = f(z, x)\left[\Delta\tau_j\right] \tag{1}$$

where $\left[\Delta\tau_j\right]$ is a translation shift sub-window operator, which can ensure effective training and inference.

2) The contraction part has inherent limitations on structural symmetry, $f\left(z, x'\right) = f\left(x', z\right)$, which is suitable for similarity learning.

B. FP AND RPN

Feature Pyramid(FP) was proposed in FPN [22] network, which can solve the problem of similar semantic goals well, because it utilizes the context information (high-level semantic information) after top-down model. For similar semantic feature targets, FPN increases the resolution of feature maps (i.e. operating on larger feature maps to obtain more useful information about similar targets). This method is used in the tracking network to increase the feature semantic information extraction of different objects. The regional proposal network was first proposed in the faster R-CNN [23].

Compare to RPN, traditional proposal extraction methods were time-consuming. For example, selective search [24] takes two seconds to process an image. Not only that, but the recommendations are not enough to test. The enumeration of multiple anchor points and the shared convolution feature enable the proposal extraction method to achieve high quality and time efficiency. RPN is able to extract more accurate proposals due to foreground - background classification and border box regression monitoring. There are several fast R-CNN variants with RPN. R-FCN [25] takes into account the location information of components. Compared with two-stage detectors, improved RPN versions such as SSD [26] and YOLO9000 [27] are effective detectors. Because of its high speed and excellent performance, RPN has many successful applications in detection and feasibility in tracking.

C. ONE-SHOT LEARNING

In recent years, one-shot learning in deep learning has attracted more and more attention. One-shot learning is used in face recognition in the early stage, by training a similarity function, to achieve the detection and matching of one sample at a time. Bayesian statistical method and meta-learning method are two main methods to solve this problem. In [28], the probability model represents the object category, and the Bayesian estimation is adopted in the inference stage. The meta-learning method is to acquire the ability to learn, and to realize and control own learning.

Single detection is regarded as a discrimination task in [29]. Its purpose is to find the parameter W that minimizes the average loss $\mathcal{L}$ of the prediction function $\psi(x; W)$. It is calculated on the data set of n samples x_i and the corresponding label ℓ_i.

$$\min_{W} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\psi\left(x_i; W\right), \ell_i\right) \tag{2}$$

A meta learning process is used to learn the parameter W of predictor from a single template z. The $\left(z; W'\right)$ is maped to the feed-forward function ω of W. Make z_i become a batch of template samples, and then express the problem as follow:

$$\min_{W'} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\psi\left(x_i; \omega\left(z_i; W'\right)\right), \ell_i\right) \tag{3}$$

As mentioned above, z represents template patch, x represents detection patch, function φ of Siamese feature extraction subnet and function ζ of region recommendation subnet, and then one-time detection task can be expressed as follow:

$$\min_{W'} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\zeta\left(\varphi\left(x_i; W\right); \varphi\left(z_i; W\right)\right), \ell_i\right) \tag{4}$$

The template branch in the Siamese subnet can be reinterpret as a training parameter to predict the kernel of the local detection task, which is usually the learning of the learning process. In this interpretation, the template branch is used to embed category information into the kernel, and the detection branch performs the detection using the embedded information. During the training phase, the meta-learner does not need any supervision other than a pair of border box supervision. In the inference phase, pruning the Siamese framework leaves only the detection branch beyond the initial frame and is therefore very fast. The target patch from the first frame is sent to the template branch and pre-computed to the detection kernel.A single detection can be perform in other frames.

III. OUR NETWORK: FPSiamRPN FRAMEWORK

In this section, we describe our proposed FPSiamRPN framework, as shown in Figure 2. Similar to SiamRPN, the framework includes Siamese subnets for feature extraction and regional proposal subnets for proposal generation. In our work,the deep network ResNet50 [30] is adopted with the addition of feature pyramid extraction as the backbone. It includes two branches in the area region proposal network subnet (RPN subnet). One is responsible for forest-background classification, and the other is used to improve candidate box. A deeply separable structure(the Deep-wise cross-correlation) is adopted for classification and regression,

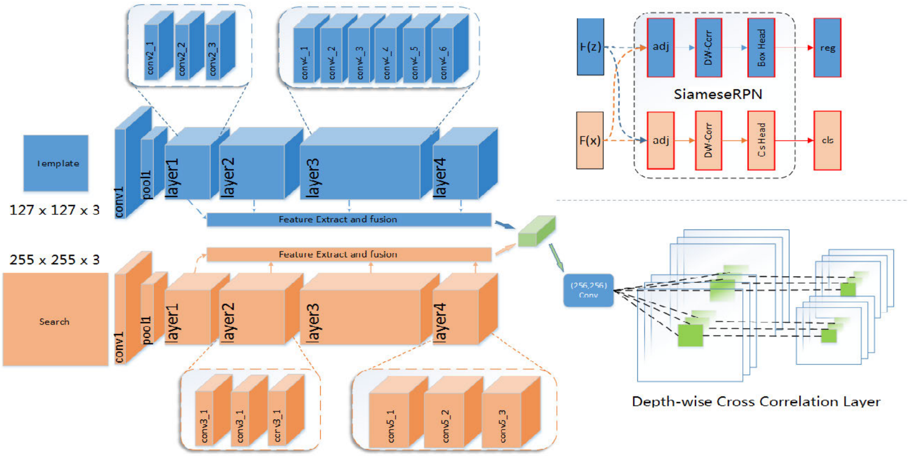

FIGURE 2. Illustration of our proposed framework. The Siamese network have two branch. One is for template and the other is for search, the outputs of the backbone sent to the RPN to get the regression and classification. The RPN architecture is shown on top right corner.

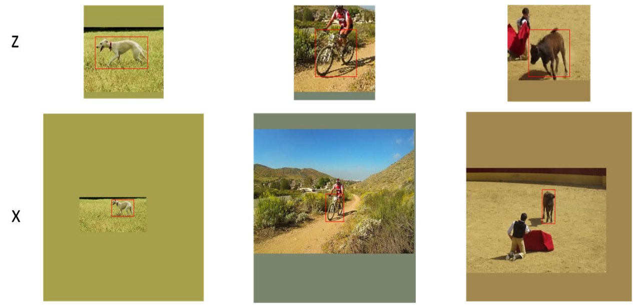

FIGURE 3. Training pairs extracted from the same video: exemplar image z and corresponding search image x from same video.

which uses 10 times fewer parameters than the original RPN network, and has the same performance.

A. OUR METHOD WITH SIAMESE NETWORK

The tracking algorithm [10], [11] based on Siamese network sets visual tracking as a cross-correlation problem and learns the similarity score map from a deep model with a Siamese network structure, in which one branch is used to learn feature representation of the target and the other is for search area. The target box is usually given in the first frame of the sequence and can be thought of as an example z. Purpose is found the most similar instances in the frame of x in embedded semantic space $\phi(\cdot)$. Feature mapping relationship is shown as follows:

$$f(z, x) = \phi(z) * \phi(x) + b \tag{5}$$

where b is the offset of the similar value. The images can be obtained from the dataset of annotated videos by extracting samples and searching for images focused on the target, as shown in Figure 3. The images are extracted from two frames of the video, both of which contain the object, which are cropped and resized to fit the input size of our network architecture. Classes that ignore objects during training. z represents the exemplar and x represents the search images. When a sub-window extends beyond image, the missing portions are filled with the mean RGB value.

The influence of center bias is droped by overcoming translability. The deep network is used for visual tracking. ResNet50 is applied as our backbone in our work. The original ResNet [30] has a 32 pixels step size, which is not suitable for dense Siamese network prediction. The original ResNet50 network is shown in Table 1.

TABLE 1. The detail data display of each layer in ResNet50 network structure. In the front in parentheses is convolution kernel size, followed by the channel number.

Layer name	outout size	ResNet50
conv1	112 x 112	7 x 7, 64, stride2
conv2_x	56 x 56	3 x 3, max pool, stride2
		[1 x 1, 64 3 x 3, 64 1 x 1, 256] x 3
conv3_x	28 x 28	[1 x 1, 128 3 x 3, 128 1 x 1, 512] x 4
conv4_x	14 x 14	[1 x 1, 256 3 x 3, 256 1 x 1, 1024] x 6
conv5_x	7 x 7	[1 x 1, 512 3 x 3, 512 1 x 1, 2048] x 3

In our work, conv4 and conv5 blocks is improved to unit space step size, the effective step length of the last two blocks is reduced from 16 pixels and 32 pixels to 8 pixels, and increasing the acceptance range by expanding convolution. Then we fuse the outputs of conv2, 3 and 4 blocks with the features of conv3, 4 and 5 up-sampling, and convoluted the channels to 256 by 1 × 1 convolution, as shown in Figure 4. In addition, we find out that careful fine-tuning of ResNet will improve performance. By setting the learning rate of the ResNet extractor 10 times smaller than the regional RPN

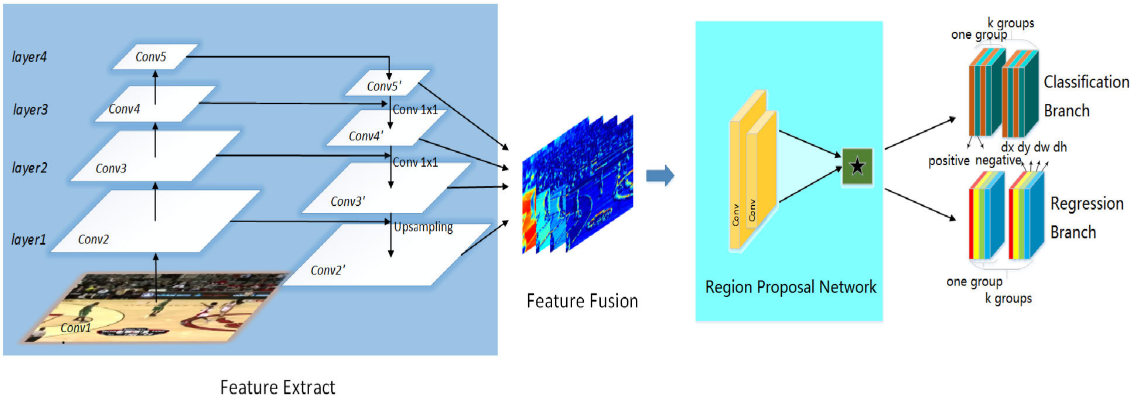

FIGURE 4. The detail description of our method and how we fuse the feature map in different layers.

portion, the feature representation can be made more suitable for tracking tasks.

B. A NOVEL LOSS TERM FOR DISTINGUISHING SIMILAR OBJECTS

We use depth-wise cross correlation to obtain the classification and regression channels. For k anchors, the network needs to output 2k channels for classification and 4k channels for regression. The goal of our learning algorithm is to train discriminant feature embedding applicable to multiple objects of the same category. The existing Siamese series of networks cannot extract the deeper semantic features well. Although DaSiamRPN [15]reduces the effects of similar distractors by proposing a distractor recognition model and uses NMS on box-selecting,if objects are too closed to each other, only the box with the highest score will be retained and all the boxes around will be discarded by using NMS to discard the box. Therefore, when lots of distractors are close to the tracking object, most of the distractors will be lost and the influence of distractors on target cannot be eliminated well.

At the present stage of tracking, most target tracking algorithms can't distinguish distractors and target well, which becomes a big problem in target tracking. In order to reduce the influence of distractors on target tracking, a discriminant example is proposed to distinguish the embedded loss of similar objects. Firstly, cross-correlate the template branch p of the Siamese subnet with the search branch z to get the score of target, which is represented by $s(p, z)$. Then, m anchors are generated around the target in the search branch z, and calculated the scores of all anchor areas d with the search area z, as $\sum_{i=1}^{N} s(d_i, z)$, sending the output characteristics into softmax function for binary classification, it determines the classification of tracking target with the surrounding objects. The proposed formula is descripted as follow:

$$\sigma_{inst} = \frac{exp\left(s\left(p, z\right)\right)}{\sum_{i=1}^{m} exp\left(s\left(d_i, z\right)\right)} \tag{6}$$

In which $\sigma_{inst}(\cdot)$ is used to compare the positive score of the tracking target with all the resulting anchor objects (including target object). According to the definition of softmax, it shows that the bigger the value of $\sigma_{inst}(\cdot)$, the greater the probability of being a target. For all the data with batch N, the following discriminating instance embedding loss is proposed, in which θ is hyperparameter, for smooth the loss in training stage:

$$\mathcal{L}_{inst} = \frac{1}{N}\sum_{i=1}^{N} \log\left(\sigma_{inst}\left(s\right) + \theta\right) \tag{7}$$

From the formula, the value in the log is the softmax value correctly classified by this group of data. So we need to small the loss of this sample to make the softmax value larger.

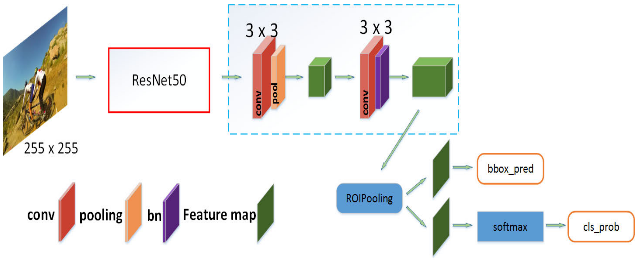

FIGURE 5. The lightweight network structure. The purpose is to extract the different detection boxes on the image, in which ResNet50 comes from the search branch x in Siamese network. The convolution kernel size is 3 × 3, and we use padding as 1 to ensure that the output feature map size is consistent with the input.

1) ANCHOR GENERATION DETAILS

It has been proved that the candidate objects obtained from the image frame is actually the object detection to the object in the image. A lightweight network is proposed to extract the distractors on target which is similar to the target detection, for generating anchors and selecting the candidate box. Our network structure consists of two convolution layers, a pooling layer and a batch normalization layer. The kernel size used in the convolution layer is 3 × 3, padding sets as 1, and the stride sets as 1. The proportion of anchor adopts the proportion of anchor in RPN. The proposed network structure is shown as Figure 5.

By inputting the feature map into the lightweight detection network, the bounding boxes of distractors on the image are generated, then calculate the proposed cross-correlation score $\sum_{i=1}^{N} s\left(d_i, z\right)$. It implements the embedding of unique features of the tracking target and can effectively distinguish similar objects that may appear around the tracking target.

For classification and regression, two branches $[\phi(z)]_{cls}$ and $[\phi(z)]_{reg}$ is used through the search area in the z channel, which do the corresponding convolution operation with two branches $[\phi(x)]_{cls}$ and $[\phi(x)]_{reg}$ in template image x. At last, obtaining the classification score is the dimension $w \times h \times 2k$ and regression score is the dimension $w \times h \times 4k$. For classified loss function, cross entropy loss function $\mathcal{L}_{cls}$ is used as follows.

$$\mathcal{L}_{cls} = -\left[y \log y' + (1-y)\log\left(1-y'\right)\right] \tag{8}$$

where y represents the ground truth, y' represents the estimate value. We use $\left\{A_x, A_y, A_w, A_h\right\}$ to represent center point and the shape of the anchor, and get $\{\delta[0], \delta[1], \delta[2], \delta[3]\}$ through normalization. When using multiple anchor points to train the network, we still use the smooth $\mathcal{L}1$ loss and regression normalization coordinates, which are shown as follows:

$$smooth_{\mathcal{L}1} = \begin{cases} 0.5\sigma^2 x^2, & |x| < \dfrac{1}{\sigma^2} \\ |x| - \dfrac{1}{2\sigma^2}, & |x| \geq \dfrac{1}{\sigma^2} \end{cases} \tag{9}$$

Finally, the loss function is optimized as follows:

$$loss = \mathcal{L}_{cls} + \lambda\mathcal{L}_{reg} + \alpha\mathcal{L}_{inst} \tag{10}$$

where λ, α are hyperparameter to balance the three parts, and $\mathcal{L}_{reg}$ is calculated as follows:

$$\mathcal{L}_{reg} = \sum_{i=0}^{3} smooth_{\mathcal{L}1}\left(\delta[i], \sigma\right) \tag{11}$$

C. ONLINE TRACK

The output of the template branch is used as the weight to track the subsequent frames. The two kernels generated in the template branch are pre-calculated on the initial frame and fixed during the whole tracking period. In the detection frame, we obtain the classification and regression output from the previous propagation, and generate multiple candidate frames.In our work,we also use SiamRPN [14] to extracte candidate boxes. Meanwhile, sine window is used and proportional change penalty to rearrange the scores of candidate boxes to get the best score. After the abnormal value is lost, adding cosine serial port can restrain large displacement.The proposed penalty item, which control the size, and proportion change are descripted as follows:

$$penalty = e^{k*\max\left(\frac{x}{x'}, \frac{x'}{x}\right)*\max\left(\frac{s}{s'}, \frac{s'}{s},\right)} \tag{12}$$

where k is a hyperparameter, x represents ratio between the height and width of the proposal, and x' represents the ratio of the last frame. s and s' represent the overall size of the proposal and the last frame, which are calculated as (12), and the s is calculated as follow:

$$s = \sqrt{(w+p) \times (h+p)} \tag{13}$$

where w and h represent the width and height of the target, p is the padding, the value is $(w+h)/2$. After that, the classification score is multiplied by the time penalty, the first k candidate boxes are reordered, and then non maximum suppression is performed to obtain the final tracking boundary box. After the final bounding box is selected, the target size is updated by linear.

IV. EXPERIMENT

A. EXPERIMENTAL DETAILS

1) TRAINING

The backbone of our architecture is pre trained image tags on ImageNet [31]. We train the network on the training set of COCO [32], ImageNet det [31], and VID [31] datasets. The training set size is more than 150GB. In training and testing,a single scale image representation template is used with 127 pixels and 255 pixels for the search area. After using ImageNet make pre train the Siamese subnet. The random gradient descent (SGD) optimizer is used to train FPSiamRPN end-to-end. Some data enhancement is used to train the regression branch,such as affine transformation.

The same object in two adjacent frames does not change much.We select fewer anchors in the tracking task than in the detection task. Therefore, only one scale of anchors with different proportions is used. In our experiment, the value of anchoring ratio is set [0.33, 0.5, 1, 2, 3].

The strategy of selecting positive and negative training samples is also very important in our framework. Here we use the criteria used in the object detection task. In our work, IoU and two thresholds $[th]_{hi}$ and $[th]_{lo}$ is used as the measurement. Positive samples are defined as anchors with IoU $> [th]_{hi}$ and its corresponding basic facts. Negative numbers are defined as anchors that satisfy IoU $< [th]_{lo}$. The parameter $[th]_{lo}$ is set to 0.3 and $[th]_{hi}$ is set to 0.6. We also set up a training pair of up to 16 negative samples and a total of 64 samples. Our experiments are implemented using PyTorch on a PC with an Intel i7, 8G RAM, NVidia GTX 2080ti.

2) EVALUATION

We focus on short-term single target tracking of OTB2015 [33], VOT2016 [34] and VOT 2018 [35]. Each dataset has 60 videos, and OTB2015 has 100 videos. All the tracking results use the reported results to ensure a fair comparison.

B. ABLATION EXPERIMENTS

1) BACKBONE ARCHITECTURE

The choice of feature extractor is important as the number of parameters and types of layers directly affect memory, speed, and performance of the tracker. We compare different network architectures for the visual tracking.

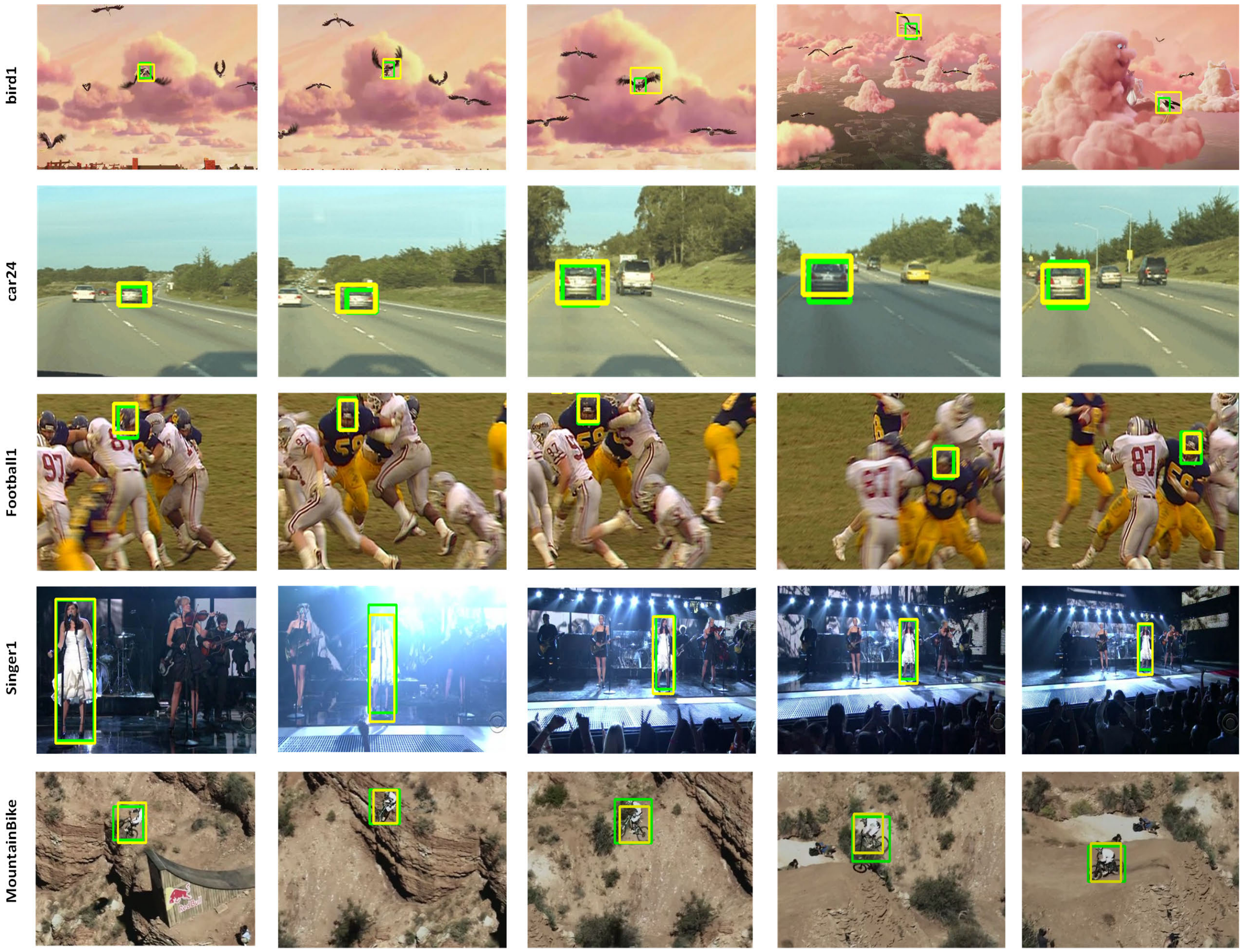

FIGURE 6. Further qualitative results of our method on sequences from the visual object tracking benchmark OTB2015. Green box represents the ground-truth and the yellow box represents our track box.

In our work, AlexNet, ResNet18, ResNet34, ResNet50, and ResNetFPN(our backbone) are used as backbones. We report performance by Area Under Curve (AUC) of success plot on OTB2015 with respect to the leading accuracy on ImageNet.

Table 2 illustrates that by replacing AlexNet to our backbone, the performance improves a lot on VOT2018 dataset. Besides, experimental results show that finetuning the backbone part is critical, which yields a great improvement on tracking performance.

2) PYRAMID FEATURE AGGREGATION

To investigate the impact of pyramid feature aggregation, we first train three variants with single RPN on ResNet50. We empirically found that conv4 in ResNet50 alone can achieve a competitive performance with 0.344 in EAO. Compare to pyramid feature aggregation(combine L3, L4, L5), pyramid feature aggregation yields a 0.363 EAO score on VOT2018, which is 7.7% higher than that of the single layer baseline.

TABLE 2. Ablation study of the proposed tracker on VOT2018 and OTB2015. L3, L4, L5 represent conv3,conv4,conv5, respectively. Finetune represents whether the backbone is trained offline. UP/DW means up channel correlation and depthwise correlation.

Backbone	L3 L4 L5	Finetune	Corr	VOT2018	OTB2015
AlexNet			UP	0.313	0.633
			DW	0.322	0.641
ResNetFPN	√ √ √		UP	0.335	0.621
	√ √ √	√	UP	0.342	0.633
ResNet-50	√ − −	√	DW	0.332	0.643
	− √ −	√	DW	0.344	0.648
	− − √	√	DW	0.325	0.639
	√ √ −	√	DW	0.337	0.651
	√ − √	√	DW	0.335	0.646
	− √ √	√	DW	0.354	0.658
ResNetFPN(ours)	√ √ √		DW	0.355	0.654
	√ √ √	√	DW	0.363	0.662

3) DEPTHWISE CORRELATION

We compare the original up-channel cross correlation layer with the proposed depthwise cross correlation layer.

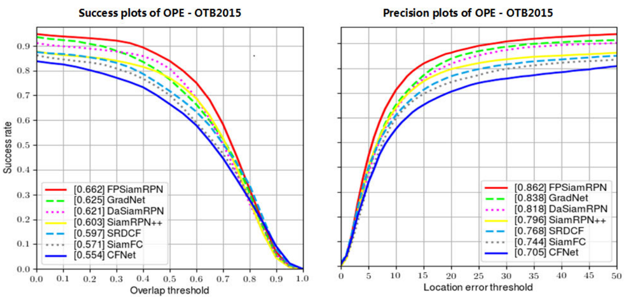

FIGURE 7. Success and precision plots show a comparison of our tracker with state-of-the-art trackers(GradNet,DaSiamRPN,SiamRPN++,SRDCF, SiamFc,CFNet) on the OTB2015 dataset.

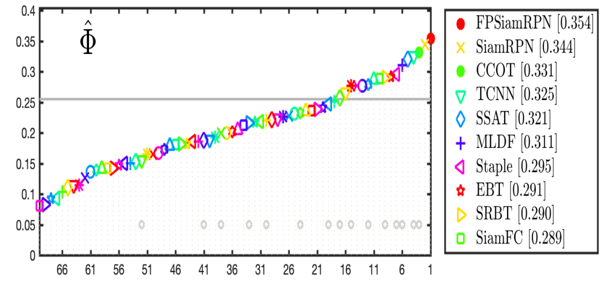

FIGURE 8. The expected average overlap(EAO) score in VOT2016. The proposed FPSiamRPN method campare with SoamRPN, CCOT, TCBB, SSAT, MLDF, Staple, EBT, SRBT, and SiamFC. Large value is better.

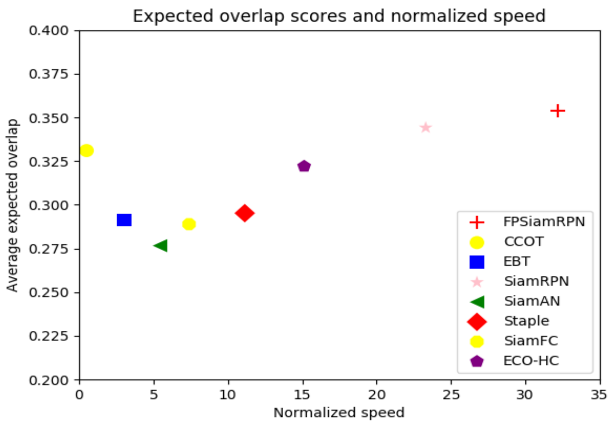

FIGURE 9. Performance and speed of our tracker and some state-of-the-art trackers in VOT2016. More closed to top means higher precision, and more closed to right means faster. FPSiamRPN is able to rank 1st in EAO.

As shown in the Table 2, the proposed depthwise correlation gains 2.2% improvement on VOT2018 and 1.2% improvement on OTB2015, which demonstrates the importance of depthwise correlation. This is partly beacause a balanced parameter distribution of the two branches makes the learning process more stable, and converges better.

C. RESULTS ON OTB2015

OTB2015 [33] contains 100 video sequences for tracking, and has been very perfect and authoritative. The evaluation results mainly rely on two indicators: accuracy and success rate. The precision plot shows the percentage of frames that the tracking results are within 20 pixels from the target. The success plot shows the ratios of successful frames when the threshold varies from 0 to 1, where a successful frame means its overlap is larger than given threshold. The area under curve of success plot is used to rank tracking algorithm.

TABLE 3. Detail information about several published state-of-art trackers' performances in VOT2016. *Red*, *blue* and *green* represent 1st, 2nd and 3rd respectively.

Tacker	EAO ↑	Accuracy ↑	Failure ↓	EFO ↑
SiamRPN	0.3441	0.56	1.08	23.3
C-COT [5]	0.331	0.53	0.85	0.507
Staple [36]	0.2952	0.54	1.35	11.14
EBT [37]	0.2913	0.47	0.9	3.011
SiamFC	0.2889	0.53	0.87	7.395
SiamRN	0.2766	0.55	1.37	5.44
FPSiamRPN	**0.354**	**0.609**	**0.67**	**32.2**

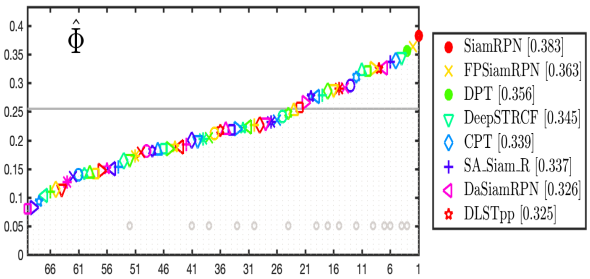

FIGURE 10. The expected average overlap(EAO) score in VOT2018. The proposed FPSiamRPN method campare with SiamRPN, DPT, DeepSTR, CPT, SA_Siam_R, DaSiamRPN, DLSTpp. Large value is better.

The standardized OTB benchmark provides a fair and robust testing platform. The Siamese based tracker formulate the tracking as a one-shot detection task without any online update, so its performance is inferior on this benchmark without resetting. However, the limited representation of shallow networks is the main obstacle to the Siamese tracker from exceeding the top-performing method, such as SiamFC [11].

In the experiment, we compared our method with a series of related tracking methods. Qualitative results of FPSiamRPN for OTB2015 sequences are shown in Figure 6. As shown in the Figure 7,comepare to the GradNet [38], SiamRPN++ [16], DaSiamRPN [15], SRDCF [39], SiamFC [11], CFNet [13], FPSiamRPN can be ranked at the top in success plot and precision plot. (The result of SiamRPN++ is trained by using our training datasets). From Figure 7, the proposed algorithm can achieve high accuracy and success rate.

D. RESULTS ON VOT2016

VOT2016 [34] dataset consists of 60 sequences. Performance is evaluated based on accuracy (the average overlap at successful tracking) and robustness failure times. The expected average overlap (EAO) is used to evaluate the

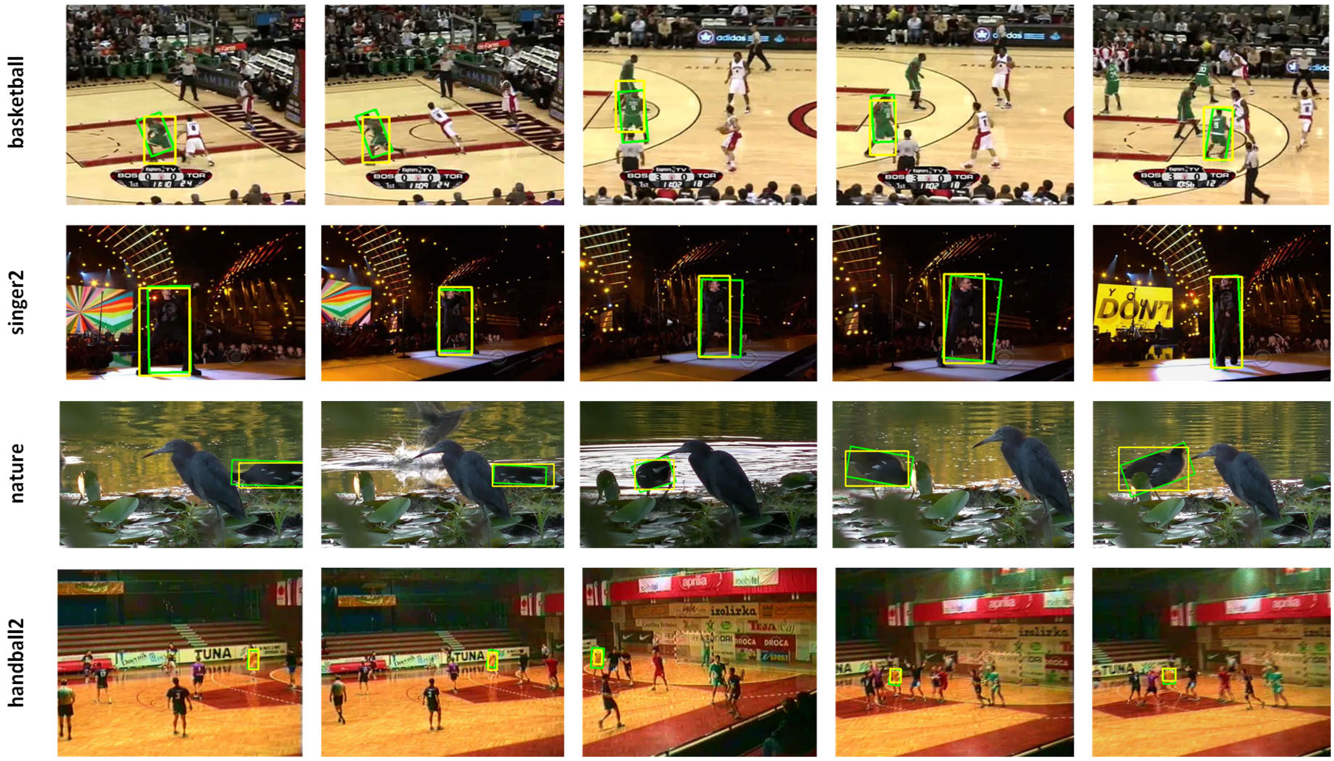

FIGURE 11. Further qualitative results of our method on sequences from the visual object tracking benchmark VOT2016. Green box represents the ground-truth and the yellow box represents our track box.

TABLE 4. Comparison with the state-of-art in terms of expected average overlap (EAO), robustness and accuracy on the VOT2018 benchmark. Our trackers obtains a well performance among the top-ranked methods.*Red, blue* and *green* represent 1st, 2nd and 3rd respectively.

	DLSTpp	DaSiamRPN	SA_Siam_R	CPT	DeepSTRCF	SiamRPN	DRT	**Ours**
EAO ↑	0.325	0.326	0.337	0.339	0.345	0.383	0.356	**0.363**
Accuracy ↑	0.543	0.569	0.566	0.506	0.523	0.586	0.519	**0.596**
Robustness ↓	0.224	0.337	0.258	0.239	0.215	0.276	0.201	**0.302**

overall performance, which considers two kinds of precision. Besides, the speed is evaluated with a normalized speed (EFO).And we compared some published state-of-art trackers, Figure 8 illustrates the EAO ranking. And further, the results of detail information about several published state-of-art trackers' performances in VOT2016 are shown in Table 3. In order to show our tracker can achieve a superior performance when operating at high speed. Figure 9 shows the performance and speed of the state-of-the-art trackers. Qualitative results of FPSiamRPN for VOT2016 sequences are shown in Figure 11. In our work,most of the sequences are ones with the distractors.

E. RESULTS ON VOT2018 AND VOT2019

We test our FPSiamRPN tracker on VOT2018 dataset [35] in comparison with some state-of-the-art methods. VOT2018 dataset is one of the most recent datasets for evaluating online model-free single object trackers, and includes 60 public sequences with different challenging factors. Following the evaluation protocol of VOT2018, the expected average overlap (EAO), accuracy and robustness are adopted to compare different trackers. The detailed comparisons are reported in Table 4. Figure 10 illustrates the EAO ranking in VOT2018. From Figure 10,in some cases, the proposed FPSiamRPN method is rank the second best campare with some state-of-the art methods. However, experimental results of the proposed method is very good in most cases.

From Table 4, the proposed FPSiamRPN method achieves the top-ranked performance on accuracy and achieves the second-ranked performance on the expected average overlap criteria. Especially, our FPSiamRPN tracker outperforms all existing trackers. Our tracker achieves a substantial improvement over the tracker(SiamRPN) with a gain of 1.7% in accuracy. Qualitative results of FPSiamRPN for VOT2018 sequences are shown in Figure 12.

In our work, the proposed method focus on reducing the impact of distractors on the target object in target tracking, and ignore some detailed features of the target. Some exerimental results from benchmark data set on VOT2019 [46], the performance of EAO, accuracy, and failure not very good.

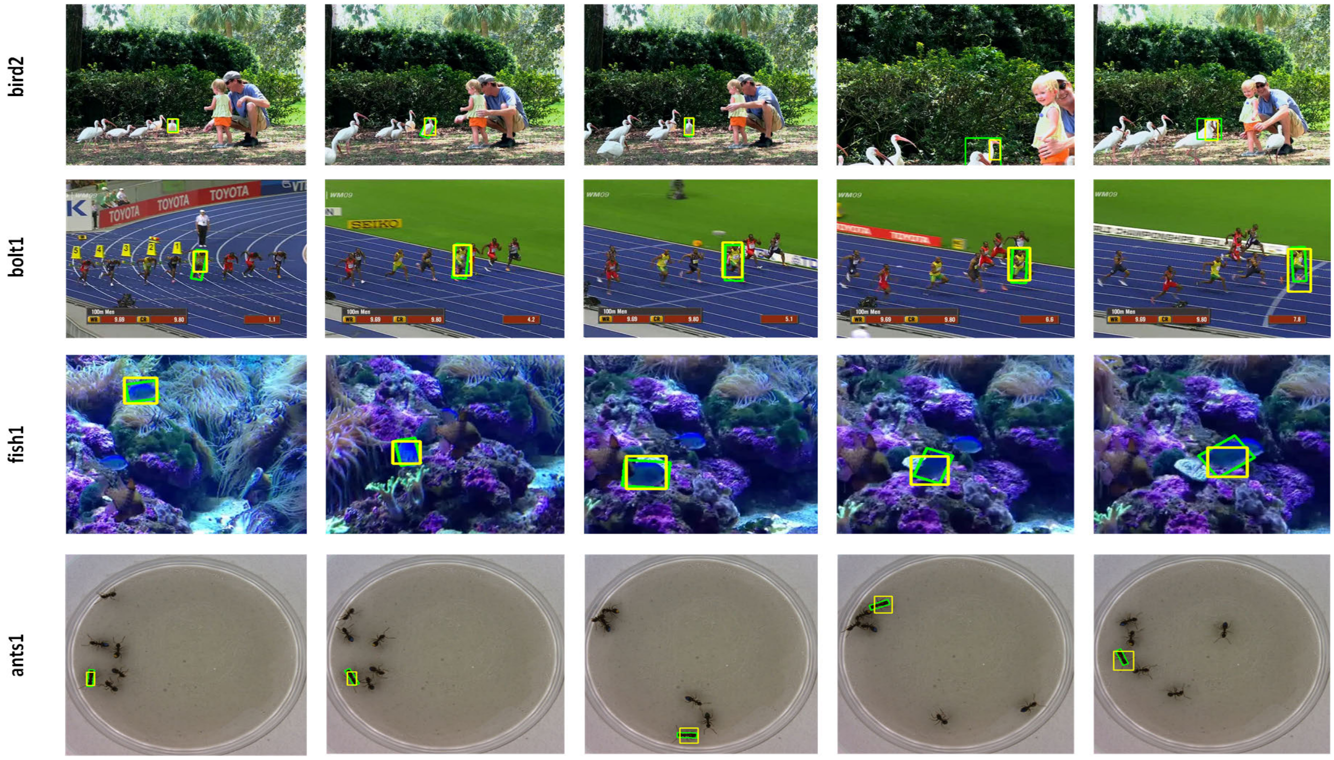

FIGURE 12. Further qualitative results of our method on sequences from the visual object tracking benchmark VOT2018. Green box represents the ground-truth and the yellow box represents our track box.

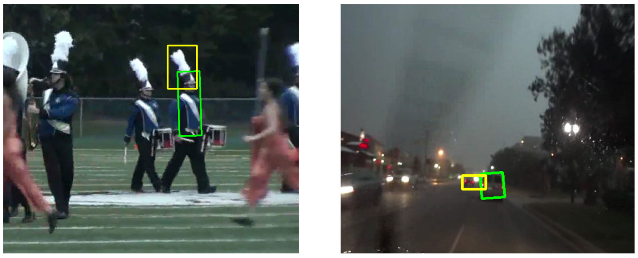

FIGURE 13. The above is the failed scene of tracking in dataset VOT2019. Our method is not enough to distinguish multiple targets in the blurred line of sight, and there is still some drift problem in distinguishing objects with high similarity.

TABLE 5. Detail information about several published state-of-art trackers' performances in VOT2019. *Red, blue* and *green* represent 1st, 2nd and 3rd respectively.

Tacker	EAO ↑	Accuracy ↑	Failure ↓
SiamRPN++ [16]	0.285	0.599	0.482
SA_Siam_R [46]	0.252	0.563	0.507
SiamCRF_RT [46]	0.262	0.549	0.346
SPM [45]	0.275	0.577	0.507
SiamBAN [40]	0.327	0.602	0.396
Ours	**0.283**	**0.578**	**0.567**

The experimental results are shown as Table 5. From Table 5, our EAO and accuracy is lower than the traditional methods.

V. CONCLUSION

In this paper, the Siamese region proposal network based on hierarchical pyramid feature fusion (FPSiamRPN) is proposed, which is end-to-end offline trained with large-scale image pairs from CoCo and ImageNet. FPSiamRPN can automatic adjust the bounding boxes and get more accurate proposal by applying box refinement procedure. In the tracking stage, one-shot detection as the reference is used. In experiment part, our method can achieve beautiful robust and good performance in OTB2015, VOT2016 and VOT2018 real-time challengers with high speed.

Due to selection box on the target detection adopt the principle of distinguishing the target and the distractors, the proposed method still has some limitations. Such as: if the selection of candidate box doesn't accurate,which will affect the calculation of the target probability with softmax. Meanwhie, the learning of the target features may be more generalized. In our work, we show some failure situations. Please see failure results in Figure 13. From Figure 13, the proposed method is not enough to distinguish multiple targets in the blurred line of sight. The results show still some drift problem in distinguishing objects with high similarity.

REFERENCES

[1] J. Xing, H. Ai, and S. Lao, "Multiple human tracking based on multi-view upper-body detection and discriminative learning," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1698–1701.

[2] L. Liu, J. Xing, H. Ai, and X. Ruan, "Hand posture recognition using finger geometric feature," in *Proc. ICIP*, 2012, pp. 565–568.

[3] G. Zhang and P. A. Vela, "Good features to track for visual SLAM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1373–1382.

[4] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[5] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[7] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[8] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Aug. 2015.

[10] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1420–1429.

[11] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 850–865.

[12] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "DCFNet: Discriminant correlation filters network for visual tracking," 2017, *arXiv:1704.04057*. [Online]. Available: http://arxiv.org/abs/1704.04057

[13] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-End representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.

[14] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[15] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 101–117.

[16] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[17] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[18] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4600.

[19] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," 2018, *arXiv:1811.07628*. [Online]. Available: http://arxiv.org/abs/1811.07628

[20] D. Gordon, A. Farhadi, and D. Fox, "Re^3: Re al-time recurrent regression networks for visual tracking of generic objects," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 788–795, Apr. 2018.

[21] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 749–765.

[22] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[24] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[25] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 379–387.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[27] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[28] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[29] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," 2016, *arXiv:1606.05233*. [Online]. Available: http://arxiv.org/abs/1606.05233

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] 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," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dolla'r, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[33] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[34] A. Kristan *et al.*, "The visual object tracking vot2016 challenge results," in *Proc. ECCV Workshops*, 2016, pp. 777–823.

[35] A. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," in *Proc. Comput. Vis. (ECCV) Workshops*. Cham, Switzerland: Springer, 2016, pp. 777–823.

[36] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.

[37] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 943–951.

[38] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6162–6171.

[39] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.

[40] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Washington, DC, USA, Jun. 2020, pp. 14–19.

[41] Q. Zhou, B. Zhong, X. Lan, G. Sun, Y. Zhang, B. Zhang, and R. Ji, "Fine-grained spatial alignment model for person re-identification with focal triplet loss," *IEEE Trans. Image Process.*, vol. 29, pp. 7578–7589, Jun. 2020, doi: 10.1109/TIP.2020.3004267.

[42] B. Zhong, B. Bai, J. Li, Y. Zhang, and Y. Fu, "Hierarchical tracking by reinforcement learning-based searching and Coarse-to-Fine verifying," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2331–2341, May 2019.

[43] Q. Zhou, B. Zhong, Y. Zhang, J. Li, and Y. Fu, "Deep alignment network based multi-person tracking with occlusion and motion reasoning," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1183–1194, May 2019.

[44] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2020, pp. 15–20.

[45] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "SPM-tracker: Series-parallel matching for real-time visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3643–3652.

[46] M. Kristan *et al.*, "The seventh visual object tracking VOT2019 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Seoul, South Korea, Oct. 2019, pp. 27–28.

YUNBO RAO (Member, IEEE) received the B.S. degree from Sichuan Normal University, in 2003, and the M.E. and Ph.D. degrees from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, in 2006 and 2012, respectively. From 2009 to 2011, he was a Visiting Scholar of electrical engineering with the University of Washington, Seattle, WA, USA. Since 2012, he has been with the School of Information and Software Engineering, University of Electronic Science and Technology of China, where he is currently an Associate Professor. His research interests include image segmentation, 3-D reconstruction, video enhancement, and medical image processing.

YIMING CHENG received the B.S. degree in the academic of software engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018, where he is currently pursuing the master's degree in software engineering. His research interests include computer vision, machine learning, and target tracking.

JUNMIN XUE received the master's degree in computer applications from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004. He is currently pursuing the Ph.D. degree in electronic and information engineering with the University of Electronic Science and Technology of China, Chengdu, China. From 2004 to 2011, he was a Team Manager with the Business and Operation Support System Design and Operation, Beijing Branch, China Unicom. He is also working as a Senior Engineer in information security with the Postal Saving Bank of China. His research interests include object detection and image segmentation.

JIANSU PU received the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2013. He is currently an Associate Professor with the Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include information visualization, visual analysis in spatiotemporal data, time series, and social networks.

QIUJIE WANG received the B.Sc. degree from the Nanjing University of Information Science and Technology (NUIST), in 1997 and the M.Sc. degree from the Guangdong University of Technology (GDUT), in 2000, in computer science and technology. She is currently a Lecturer with GDUT. Her current research interests include network information security, intelligent video processing, big data applications, and security.

RIZE JIN received the M.S. and Ph.D. degrees in computer engineering from Ajou University (AU), South Korea, in February 2011 and February 2015, respectively. He worked as a Postdoctoral Researcher with the Department of Computer Engineering, Korea Advanced Institute of Science and Technology (KAIST), South Korea. He was an Assistant Professor with the Software Department, AU. He is currently a Professor with the School of Computer Science and Technology, Tianjin Polytechnic University, China. His research interests include flash-based DB, NoSQL, natural language processing, and deep learning.

QIFEI WANG (Member, IEEE) received the B.S. degree in information and computing science from the Beijing University of Posts and Telecommunications, China, in 2007, and the Ph.D. degree in control science and engineering from Tsinghua University, China, in 2013. He joined EECS, University of California at Berkeley, Berkeley, CA, USA, in 2014. His current research interests include computer vision, machine learning, video processing, and communications.