electronics

MDPI

Article

ACSiamRPN: Adaptive Context Sampling for Visual Object Tracking

Xiaofei Qin [1,2,3], Yipeng Zhang [4], Hang Chang [5], Hao Lu [6] and Xuedian Zhang [1,2,3,7,*]

[1] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; xiaofei.qin@usst.edu.cn

[2] Shanghai Key Laboratory of Contemporary Optics System, Shanghai 200093, China

[3] Key Laboratory of Biomedical Optical Technology and Devices of Ministry of Education, Shanghai 200093, China

[4] School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; 183780715@st.usst.edu.cn

[5] Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA; HChang@lbl.gov

[6] Guangxi Yuchai Machinery Co., Ltd., Nanning 530007, China; luhao@yuchai.cn

[7] Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 200092, China

* Correspondence: xdzhang@usst.edu.cn

Received: 17 August 2020; Accepted: 16 September 2020; Published: 18 September 2020

Abstract: In visual object tracking fields, the Siamese network tracker, based on the region proposal network (SiamRPN), has achieved promising tracking effects, both in speed and accuracy. However, it did not consider the relationship and differences between the long-range context information of various objects. In this paper, we add a global context block (GC block), which is lightweight and can effectively model long-range dependency, to the Siamese network part of SiamRPN so that the object tracker can better understand the tracking scene. At the same time, we propose a novel convolution module, called a cropping-inside selective kernel block (CiSK block), based on selective kernel convolution (SK convolution, a module proposed in selective kernel networks) and use it in the region proposal network (RPN) part of SiamRPN, which can adaptively adjust the size of the receptive field for different types of objects. We make two improvements to SK convolution in the CiSK block. The first improvement is that in the fusion step of SK convolution, we use both global average pooling (GAP) and global maximum pooling (GMP) to enhance global information embedding. The second improvement is that after the selection step of SK convolution, we crop out the outermost pixels of features to reduce the impact of padding operations. The experiment results show that on the OTB100 benchmark, we achieved an accuracy of 0.857 and a success rate of 0.643. On the VOT2016 and VOT2019 benchmarks, we achieved expected average overlap (EAO) scores of 0.394 and 0.240, respectively.

Keywords: visual object tracking; SiamRPN; global context; selective kernel convolution

1. Introduction

Visual object tracking is one of the most basic problems in the application of human–computer interaction, visual analysis and auxiliary drive systems. Its purpose is to accurately estimate the position and scale of the object in the subsequent frame, according to the bounding box given in the first frame [1]. The appearance difference caused by illumination, deformation, occlusion, rotation and motion is a great challenge. In addition, the tracking speed is also very important in practical application. Generally, the real-time tracking is at least 25 Frames Per Second (FPS).

Video tracking technology has developed rapidly in the past few years. In particular, the Siamese network, based on a region proposal network (SiamRPN) [2] proposed by Li et al., adds the idea of a region proposal network (RPN) [3] in object detection to the Siamese network [4] and avoids multi-scale testing, thus greatly increasing the target tracking speed. The network can run at the speed of 160 FPS. During tracking, the network directly outputs the scores of the foreground and background and the coordinates of the center point of the bounding box, as well as its width and height. SiamRPN is an important milestone in visual object tracking fields, as many works are carried out on the basis of SiamRPN. Siamese cascaded region proposal networks (C-RPNs) [5], proposed by Heng et al., regress the bounding box prediction progressively by connecting multiple RPNs in cascaded form and fusing the multi-layer features of backbone networks to improve tracking accuracy and robustness. In DaSiamRPN [6], in order to improve the tracker's generalization and discrimination abilities, the authors, in the training stage, enrich the positive sample by introducing the existing detection dataset and enrich the negative sample by introducing semantic negative pairs, consisting of labeled targets both in the same categories and different categories. In order to deal with the problem of long-term tracking, the authors propose a switching method between short-term tracking and failure cases in the inference stage. Deeper and wider Siamese networks (SiamDWs) [7] proposed the cropping inside residential (CIR) method to modify the original residual unit so that Siamese networks could take advantage of the capability of wider and deeper backbone networks, such as ResNet [8], Inception [9] and ResNeXt [10]. After that, Li et al. proposed SiamRPN++ [11], which greatly deepened the depth of the network by a simple yet effective spatially aware sampling strategy. Through multi-layer information fusion, SiamRPN++ achieved a very high tracking accuracy. Although the methods mentioned above have achieved promising accuracy, they do not consider the relationships and differences between the long-range context information of various objects. Their improvement of accuracy mainly depends on the increase of model complexity, such as network depth, width and module stack number, which inevitably leads to cumbersome models and relatively low speeds.

As a general idea of neural networks, a visual attention mechanism is widely used in computer vision tasks such as image classification, semantic segmentation, face recognition and human pose estimation among others. Its core idea is to find the relevance among different features in different tasks, and then highlight some important features, such as channels, pixels, multi-level features, and so on. SENet [12], proposed by Hu et al., collects the information of each channel by global average pooling (GAP), remodels the relationship between channels by 1 × 1 convolution, and then reassigns the weights to the original channels. The selective kernel network (SKNet) [13] proposes an adaptive selection mechanism, which divides the common convolution operation into three steps: split, fuse and select. Split operation allows different kernel size convolutions in multiple parallel branches to extract features with different receptive fields. Fuse operation sums the features of different receptive fields in an elementwise manner, and then applies GAP to generate channel-wise attentions for all receptive fields. Finally, the select operation is carried out to select the appropriate receptive fields and features [14]. SENet and SKNet introduce channel attention mechanisms; however, they do not consider spatial attention. Convolutional Block Attention Module (CBAM) [15], proposed by Sanghyun Woo et al., makes a further improvement of SENet, which uses both global maximum pooling (GMP) and GAP to generate channel attention and spatial attention. However, the spatial attention in CBAM is limited by the 7 × 7 convolution used after GAP and GMP operations, and it only can be seen as a kind of local spatial attention. Kaiming He et al. proposed a non-local network (NLNet) [16]. For each query position, NLNet first calculates the pairwise relationship between the query position and all positions to form a global spatial attention map, then computes the response at a position as a weighted sum of the features at all positions. Although NLNet can collect the global context information of an entire feature map, its calculation is extremely large. To address this issue, a global context network (GCNet) [17] creates a simplified network based on a query for independent formulation, which maintains the accuracy of NLNet, but with significantly less computation. However, NLNet and GCNet do not consider channel attention.

This paper follows the structure of SiamRPN. To solve the problem of poor adaptability of the general object tracker to tracking scenes, the global context block (GC block) is adopted in the template branch of SiamRPN, which can collect the long-range context information of an object; thus, the global spatial attention mechanism is introduced into the object tracker. In order to improve the adaptability of the object tracker to the object scales, we design a cropping-inside selective kernel block (CiSK block) based on SKNet and replace the 3 × 3 convolutions in the RPN part of SiamRPN with CiSK blocks. Due to its multi-branch structure, a CiSK block can provide SiamRPN with a dynamic receptive field ability. In addition, the GAP and GMP used in the fuse step of the CiSK block enrich the channel attention information of SiamRPN. The source code of our method is available at https://github.com/linjiangxiaoxian/ACSiamRPN.

2. Methods

In this section, we will describe in detail the ACSiamRPN framework that we propose for single object tracking. As shown in Figure 1, ACSiamRPN includes a Siamese subnet for feature extraction and an RPN subnet for bounding box prediction. There are two branches in the RPN subnet: one is responsible for foreground and background classification, and the other is for proposal refinement. The whole framework can be trained end to end. The ACSiamRPN framework is modified from the original SiamRPN by using a GC block and a CiSK block. The GC block can extract global context information and facilitate subsequent processing. The CiSK block has a dynamic receptive field, and the cropping operation added to the CiSK can alleviate the negative impact of padding to object localization. The four CiSK blocks in the RPN subnet have the same structure, but do not share weight. The channel numbers of features outputted by the CiSK blocks are kept at 256. Then, the final output is obtained through cross-correlation and 1 × 1 convolution. Details of the GC block and CiSK block will be described in the following part of this section.

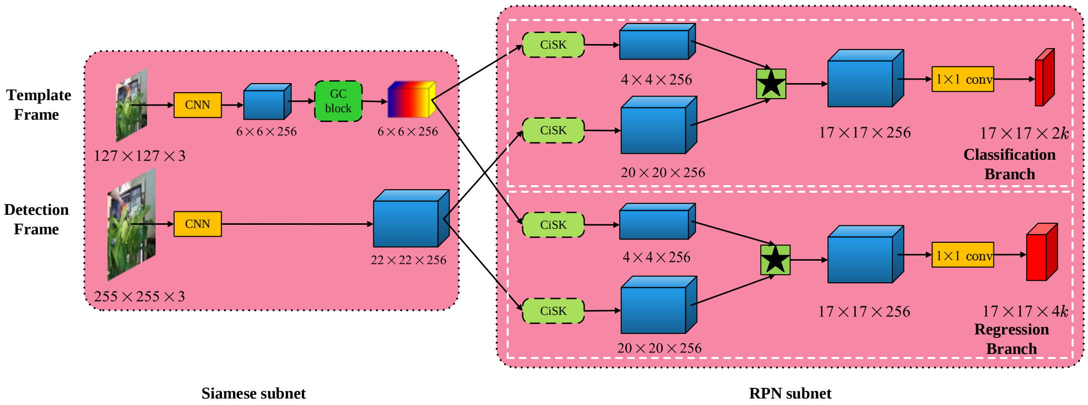

Figure 1. The architecture of ACSiamRPN. Video frames are input to a Siamese subnet to extract features. The extracted features are used as input by a subsequent RPN subnet to predict the object bounding box. The output of classification branch of the RPN subnet has *2 k* channels, representing the foreground and background scores of *k* anchors, respectively. The output of the regression branch of the RPN subnet has *4 k* channels, representing the four correction offsets for the predicted bounding box of *k* anchors, which are the correction offsets for the horizontal and vertical coordinates of the bounding box center and the correction offsets for the width and height of the bounding box.

2.1. Global Context Block

In order to model the long-range context of the template frame and deepen the network's global understanding of the current tracking scene, we added a GC block [17] to the template branch of the original DaSiamRPN. Please note that long-range context here is not a temporal concept, but a spatial one. Long-range context means the relationship between pixels that are far away from each other in the same frame. As shown in Figure 2, the GC block is composed of two parts, namely the context modeling part and the channel transforming part. Although the GC block is a lightweight block, we only added it to the template branch. The reason for this is the template branch needs to be run only once at the first frame. However, the detection branch needs to be run multiple times at all the subsequent frames, so any addition to the detection branch will affect the tracking speed. The template branch with the GC block can provide a more stable and reliable template for subsequent frames to match.

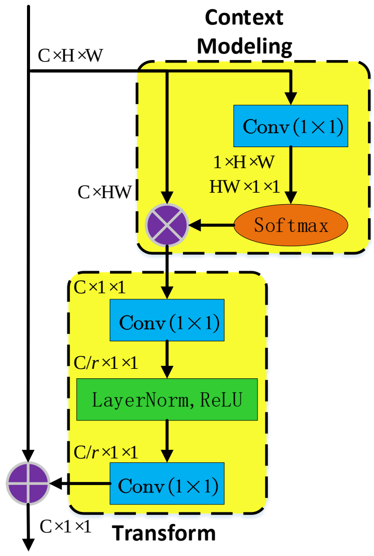

Figure 2. The architecture of the GC block. C × H × W means the channel number, height and width of the input feature map, and *r* is the channel compression ratio, where $\bigoplus$ denotes broadcast and elementwise summation, and $\bigotimes$ denotes matrix multiplication.

For the context modeling part, the main function is to establish the relationship between contexts. Conventional convolution can only catch the local context information, and the GAP and GMP operations are only simple statistical calculations, which cannot model global context well. The context modeling module groups the features of all positions together via weighted averaging to obtain the global context features [13] (it can be regarded as global attention pooling). In the context modeling part, the Channel number, Height and Width input features (C × H × W) are first convoluted by a 1 × 1 kernel, and the channel number is compressed to 1 to obtain a 1 × H × W feature map. Then, the feature map is reshaped to HW × 1 × 1 and fed into a softmax function to obtain the attention weight (HW × 1 × 1). Finally, matrix multiplication is performed between the reshaped original features (C × HW) and the attention weight (HW × 1 × 1) to get the output of the context modeling part (C × 1 × 1).

For the channel transforming part, the main function is to complete information transformation and assign the context established by the context modeling part to the corresponding channel. Similar to SENet, this part uses 1 × 1 convolution to model the relationship between channels. First, the channel number is compressed to *1/r* (in our experiment, we set *r* to 4), and then the channel number is restored to C. In this way, a bottleneck is formed, and the calculation and parameter amounts are reduced. Layer normalization is added to facilitate the training and optimizing process, and Rectified Linear Unit (ReLU) activation is used to increase the model's non-linearity.

Finally, the output C × 1 × 1 vector is broadcasted and added elementwise with the original feature map to get the final output. To summarize, the GC block can be formulated as

$$z_i = x_i + W_{v2}\mathrm{ReLU}(\mathrm{LN}(W_{v1}\sum_{j=1}^{N_p}\frac{e^{W_k x_j}}{\sum_{m=1}^{N_p} e^{W_k x_m}}x_j)) \tag{1}$$

where x and z represent the input and output of the GC block and x_j, x_i, x_m and z_i are the elements of x and z. N_p is the number of elements in x. $\frac{e^{W_k x_j}}{\sum_{m=1}^{N_p} e^{W_k x_m}}$ represents the softmax function. W_{v1} represents the weight used in the first convolution module of the channel transforming part, W_{v2} represents the weight used in the last convolution module of the channel transforming part. ReLU represents the activation function, and LN represents the layer normalization operation.

2.2. Cropping-Inside Selective Kernel Block

During object tracking, the object scale is random and may vary over time so that receptive fields have crucial influence on tracking performance. Networks such as Inception [9] have several receptive fields due to their multiple parallel branches with different kernel sizes; however, the weight of each branch is fixed in the fusion step, making it not adaptive to objects of different scales. SKNet [13] is famous for its simplicity and efficiency. SKNet can adaptively assign the weights of different branches according to the scales of different objects, making it suitable for tasks handling objects of random sizes, such as object tracking.

The proposed CiSK block as shown in Figure 3 was inspired by SKNet, which inherits its dynamic receptive field ability. In order to better apply it to object tracking tasks, we made two main improvements to SKNet. Firstly, we think that besides average pooling, maximum pooling is another important method for gathering discriminative features. Therefore, in the fuse step of SKNet, we added an extra branch starting with GMP, forming a two-branch channel attention-generating module. The GMP branch shares weights with the GAP branch. Secondly, a padding operation is necessary for SKNet due to the same convolution it used to maintain feature dimensions. However, padded values around the original feature induced potential position bias in model training [7], and thus the prediction accuracy is expected to be degraded, especially when an object moves near the search range boundary. To address this issue, the most padding-affected elements around the feature after the select step were cropped out. The detailed working process of the CiSK block is as follows.

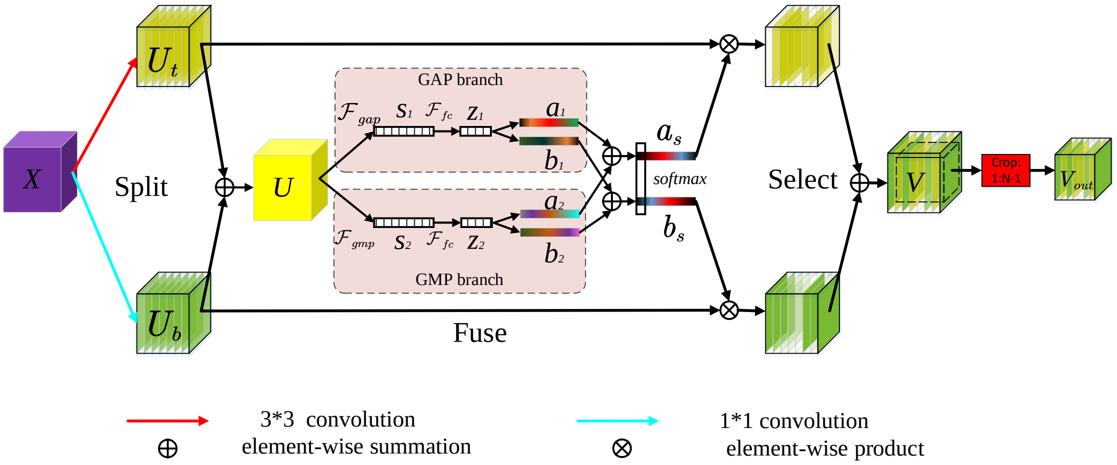

Figure 3. Our CiSK block consists of three parts: split, fuse and select.

The input feature X is convoluted by a 3×3 and 1×1 kernel respectively to obtain the top-branch feature U_t and bottom-branch feature U_b. U_t and U_b have the same spatial dimension and channel number. The size of U_t and U_b depends on which branch they source from. If it sources from the template branch, the size is $6 \times 6 \times 256$. If it sources from the search branch, it is $22 \times 22 \times 256$. Then, we add U_t and U_b element by element to get U:

$$U = U_t + U_b \tag{2}$$

The vectors s_1 and s_2 are obtained by GAP and GMP of U, where c is the cth channel of s_1 and s_2:

$$\mathrm{s}_{1c} = \mathcal{F}_{gap}(U_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} U_c(i,j) \tag{3}$$

$$\mathrm{s}_{2c} = \mathcal{F}_{gmp}(U_c) = max(U_c) \tag{4}$$

Then, the channel number of s_1 and s_2 were reduced to one-eighth of their original values by using a fully connected layer. z_1 and z_2 can be formulated as

$$z_1 = \mathcal{F}_{fc}(s_1) = \delta(\mathcal{B}(Ws_1)) \tag{5}$$

$$z_2 = \mathcal{F}_{fc}(s_2) = \delta(\mathcal{B}(Ws_2)) \tag{6}$$

where δ represents the ReLU activation function and $\mathcal{B}$ represents batch normalization.

After that, two fully connected layers were used with z_1 to recover its original dimension so a_1 and b_1 were obtained, respectively. The same operation was also done to z_2 to obtain a_2 and b_2. These processes can be formulated as

$$a_1 = Az_1,\ b_1 = Bz_1 \tag{7}$$

$$a_2 = Az_2\ ,\ b_2 = Bz_2 \tag{8}$$

where A and B are the transformation matrices when the dimension is increased.

a and b were obtained by adding a_1 and a_2 and b_1 and b_2 together, and they were normalized by a softmax function to get a_s and b_s:

$$a = a_1 + a_2,\ b = b_1 + b_2, \tag{9}$$

$$a_s = \frac{e^a}{e^a + e^b},\ b_s = \frac{e^b}{e^a + e^b}\ , \tag{10}$$

Then, a_s and b_s were multiplied by U_t and U_b with the broadcasting mechanism. The products of multiplication are summed up to get feature map V:

$$V_c = a_c \cdot U_t + b_c \cdot U_b \tag{11}$$

where $s.t.a_c + b_c = 1$.

Finally, the outermost pixels of each channel of V were cropped to get the final output V_{out}:

$$V_{out} = crop(V) \tag{12}$$

where the size of V is 6 × 6 × 256 if sourced from the template branch or 22 × 22 × 256 if sourced from the search branch. After cropping, the size of V_{out} becomes 4 × 4 × 256 or 20 × 20 × 256 correspondingly.

3. Experiments

3.1. Implementation Details

We took the AlexNet [18] that was pre-trained from ImageNet [19] as the backbone network for feature extracting and trained 20 epochs in total. First, we froze the parameters of the pre-trained AlexNet, then trained other parts for 10 epochs. After that, we unfroze the last two layers of AlexNet and trained it together with other parts of the network for 10 epochs. The total loss is the sum of the classification loss and the standard smooth L_1 loss for regression. Stochastic gradient descent (SGD) was used for optimizing, and the momentum parameter was set to 0.9. During training, the learning rate was arranged as follows. For the first 5 epochs, the learning rate increased exponentially from 0.005 to 0.01. For the remaining 15 epochs, the learning rate decreased exponentially from 0.01 to 0.0005.

During inferencing, we regarded our tracker as a local one-shot detection framework, in which the bounding box in the first frame was the only exemplar. This exemplar was sampled through the template branch only once, and the template branch was pruned after that to accelerate the tracking speed [2]. Subsequent frames were sampled by searching branches and fed into the RPN subnet to get the refined proposal. In addition, our tracker was designed for short-term object tracking, so no online template update mechanism was used.

We used four datasets as training sets, namely ImageNet VID [19], ImageNet DET [19], COCO [20] and YouTube-BB [21], and used OTB100 [22], VOT2016 [23] and VOT2019 [24] to evaluate the proposed method. Before being fed into the tracker, template frame images were resized to 127 × 127, and the search frame images were resized to 255 × 255.

Our tracker was implemented using PyTorch framework with Python on an Intel(R) Xeon(R) CPU E5-1620 v3 @3.50GHz and two NVIDIA GTX 1080Ti GPUs with 22 GB of memory in total.

3.2. Result on OTB100

We used the standard OTB100 benchmark to evaluate the performance of our tracker, which contained 100 fully annotated real-world sequences. These sequences had 11 challenges, namely illumination variation (IV), deformation (DEF), motion blur (MB), out-of-plane rotation (OPR), low resolution (LR), occlusion (OCC), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background cluttered (BC) and scale variation (SV). There were two evaluation criteria. One was the overlap rate of the bounding box and the other was the center positioning error of the bounding box.

We could get two graphs to demonstrate the performance of multiple models, those being precision plots of one-pass evaluation (OPE) based on the center positioning error of the bounding box and success plots of OPE based on the overlap rate of the bounding box. Horizontal values in these two graphs are the thresholds of those two criteria. The precision value shows the percentage of frames that meet the distance of the center point, and the success rate value shows the percentage of frames that meet the overlap rate. We compared it with nine state-of-the-art methods, including SiamRPN, MEEM [25], MUSTer [26], SiamFC [4], DSST [27], KCF [28], Struck [29], TLD [30] and CSK [31]. The results are as follows.

The number in the precision plot of OPE in Figure 4 is the precision value when the location error threshold is 20, which is the official evaluation metric used by Object Tracking Benchmark (OTB) dataset. The number in the success plots of OPE in Figure 4 is the area under the curve (AUC). As shown in Figure 4, we achieved the best results on the OTB100 benchmark, with precision 0.5 percentage points higher than the baseline (SiamRPN) and a success rate 1 percentage point higher than the baseline.

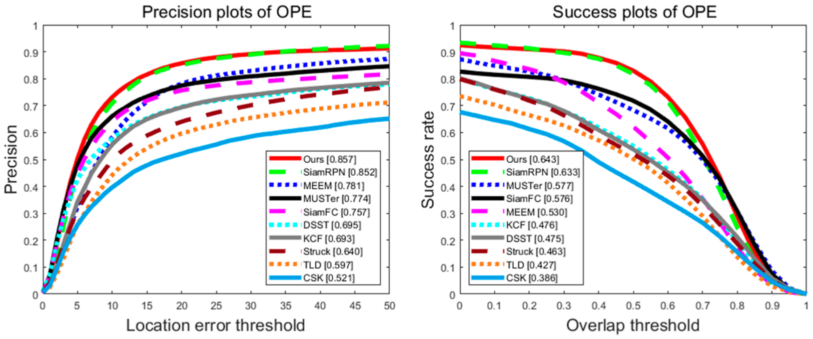

Figure 4. Success plot and precision plot of OTB100.

As shown in Figure 5, we compared ACSiamRPN with two classic trackers (SiamFC and SiamRPN) and showed the results of six videos in OTB100, which are some frames in *Skiing*, *MotorRolling*, *CarScale*, *Liquor*, *Tiger1* and *Lemming*. It can be found that when an object is too small (*Skiing*), the object rotates in plane (*MotorRolling*), the object scale changes greatly (*CarScale*) or the object is occluded (*Liquor*, *Tiger1* and *Lemming*). The two classic trackers often got inaccurate tracking results or even tracking failure. Our tracker can handle these challenges better. We think the main reasons are that, first, the GC block can collect long-range context information and improve a network's understanding of tracking scenes. Second, the CiSK block can adjust the receptive field adaptively according to the variation of object features during the tracking process, so it can better estimate the current scale of an object.

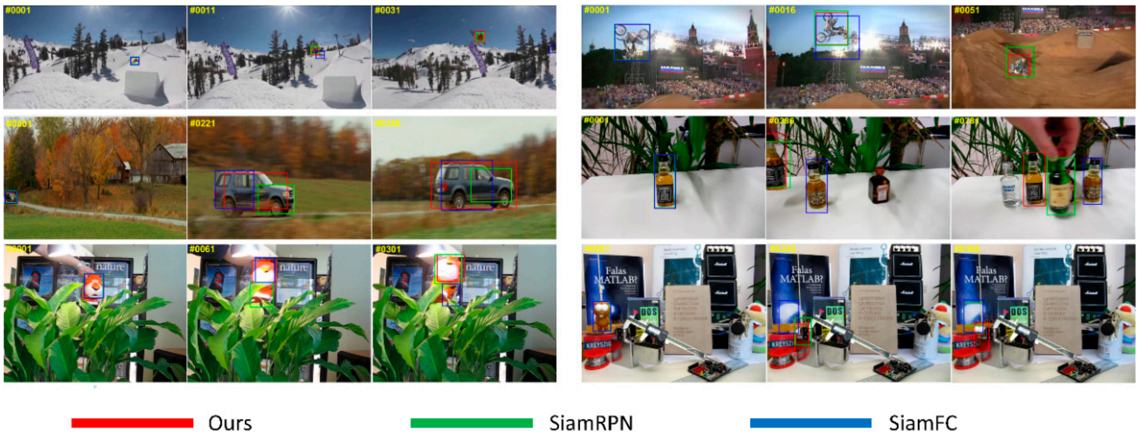

Figure 5. Tracking results of our tracker and two other trackers on six typical sequences from TB100.

3.3. Result on VOT2016

Visual Object Tracking (VOT) benchmarks evaluate a tracker by applying a reset-based methodology. Whenever a tracker has no overlap with the ground truth, the tracker will be re-initialized after five frames. The major evaluation metrics of VOT benchmarks are accuracy (A), robustness (R) and expected average overlap (EAO). An excellent tracker should have high A and EAO scores but a low R score.

We used the VOT2016 benchmark to test our tracker and compared it with nine advanced trackers. The VOT2016 public dataset was used for single object short-term tracking tasks, including 60 video sequences. We compared EAO, A and R, three criteria of the different trackers, and the details are shown in Table 1 and Figure 6.

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

Trackers	EAO	Accuracy	Robustness
Ours	0.397	0.601	0.252
SiamDW [7]	0.376	0.580	0.240
SiamRPN [2]	0.344	0.560	0.260
CCOT [32]	0.331	0.539	0.238
TCNN [33]	0.325	0.554	0.268
SSAT [23]	0.321	0.577	0.291
MLDF [34]	0.311	0.490	0.233
Staple [35]	0.295	0.544	0.378
EBT [36]	0.291	0.465	0.251
SRBT [23]	0.290	0.496	0.350

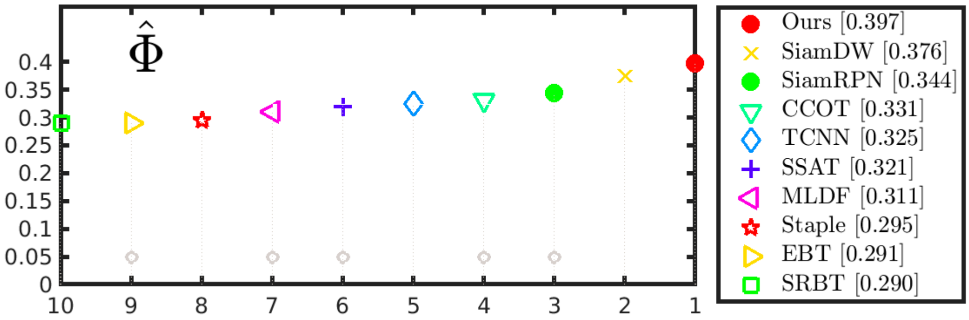

Figure 6. Expected average overlap (EAO) scores in the VOT2016 challenge. A larger value is better.

As shown in Table 1 and Figure 6, our tracker reached 0.397 EAO, 0.601 accuracy, and 0.252 robustness. Our EAO and accuracy criteria were about 15.4% and 7.3% higher than the baseline (SiamRPN), respectively, and the robustness (failure rate) was reduced by about 3%.

3.4. Result on VOT2019

We used the VOT2019 benchmark to test our tracker and compared it with nine advanced trackers. Like VOT2016, the VOT2019 public dataset was also used for single object short-term tracking tasks, and it includes 60 video sequences. Compared to VOT2018, VOT2019 replaced 20% of sequences with more difficult ones. We compared EAO, A and R, the three criteria of different trackers, and the details are shown in Table 2 and Figure 7.

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

Trackers	EAO	Accuracy	Robustness
Ours	0.240	0.562	0.642
SSRCCOT [24]	0.234	0.495	0.507
MemDTC [24]	0.228	0.485	0.587
SiamRPNX [24]	0.224	0.517	0.552
Siamfcos [24]	0.223	0.561	0.788
TADT [37]	0.207	0.516	0.677
CSRDCF [38]	0.201	0.496	0.632
CSRpp [24]	0.187	0.468	0.662
FSC2F [24]	0.185	0.480	0.752
ALTO [24]	0.182	0.358	0.818

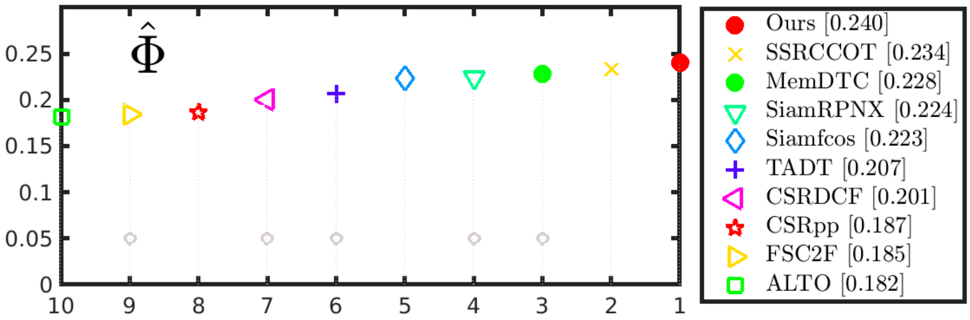

Figure 7. EAO scores in the VOT2019 challenge. A larger value is better.

As shown in Table 2 and Figure 7, our tracker was ranked first in both the EAO and accuracy criteria, while the robustness ranking was slightly behind. Among them, EAO and accuracy are about 2.6% and 13.5% higher than the second-ranked tracker, respectively.

In the VOT2019 ranking list, the performance of some trackers based on Siamese networks were better than ours, such as SiamDW_ST, SiamMask and SiamRPN++. The main reason is that they use much deeper backbone networks, such as ResNet, InceptionNet and so on, so they can extract richer target features. The backbone network used in ACSiamRPN is a five-layer AlexNet, and thus our network is relatively lightweight and can achieve a higher tracking speed. We compared the performance and tracking speed of ACSiamRPN with SiamDW_ST, SiamMask and SiamRPN++. The result is shown in Figure 8.

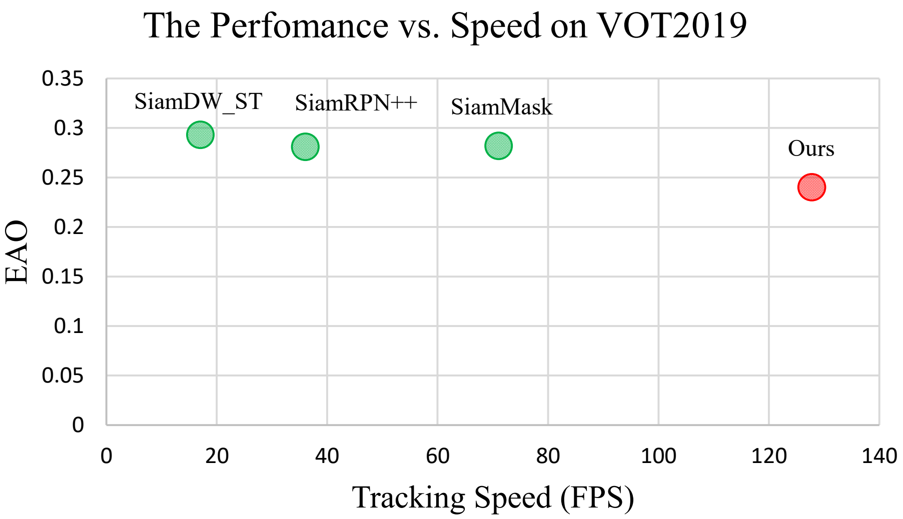

Figure 8. Comparison of the performance and tracking speeds of ACSiamRPN, SiamDW_ST, SiamMask and SiamRPN++.

As shown in Figure 8, the speed of our tracker is much higher than that of other siamese network-based networks on the VOT2019 benchmark. For example, our tracker is 57 FPS faster than SiamMask (128 vs. 71), while the EAO is only 0.042 (0.24 vs. 0.282) lower than it.

3.5. Ablation Study

The GC block and CiSK block are the two main contributions of our model. In order to study their effectiveness, we carried out ablation experiments on VOT2016. As shown in Table 3, both the GC and CiSK blocks played a positive role. Although only adding the GC block to the template branch made the network no longer symmetrical, during training, the network will adaptively adjust the weight of two branches to output features that are conductive to subsequent template matching operations. In addition, the GC block can provide a better template for the tracker. A model with the GC block alone obtained performance gains of 0.022 (0.366 vs. 0.344) in EAO and 0.037 (0.597 vs. 0.560) in accuracy. The robustness criterion was basically not affected (0.265 vs. 0.260). The model with the CiSK block alone obtained performance gains of 0.029 (0.373 vs. 0.344) in EAO and 0.039 (0.599 vs. 0.560) in accuracy. The robustness criterion was basically not affected (0.265 vs. 0.260). The model with both the GC and CiSK blocks obtained performance gains of 0.053 (0.397 vs. 0.344) in EAO, 0.041 (0.601 vs. 0.560) in accuracy, and 0.008 (0.252 vs. 0.260) in robustness.

Table 3. Effectiveness study of the global context (GC) block and cropping-inside selective kernel (CiSK) block. *Red* represents the best results.

Settings	VOT2016		
	EAO	**Accuracy**	**Robustness**
SiamRPN	0.344	0.560	0.260
SiamRPN+GC	0.366	0.597	0.265
SiamRPN+CiSK	0.373	0.599	0.270
ACSiamRPN(Ours)	0.397	0.601	0.252

EAO is a new performance criterion introduced in VOT2015, which combines the raw values of accuracy and robustness and forms a kind of hybrid criterion. EAO measures the expected no-reset overlap of a tracker run on a short-term sequence [23]. EAO has a clear practical interpretation and provides a more reasonable measure for short-term object tracking tasks, and thus it is officially recognized as the ranking criterion by the VOT competition. As shown in Table 3, the GC block and CiSK block both have obvious contributions for the improvement of the EAO, which demonstrates their effectiveness.

Cropping operation, GAP branch and GMP branch are the three main modifications in CiSK. In order to study their effectiveness, we carried out ablation experiments on VOT2016. As shown in Table 4, when GAP and GMP branches are used at the same time, the performance of the model is better than when only a GAP or GMP branch is used (0.370 vs. 0.369/0.367 when a cropping operation is not adopted, 0.397 vs. 0.382/0.395 when a cropping operation is adopted). It also can be seen that model performance gains a considerable improvement in different GAP and GMP combinations due to the cropping operation (0.382 vs. 0.369, 0.395 vs. 0.367, 0.397 vs. 0.370).

Table 4. Effectiveness study of cropping operation, global average pooling (GAP) and global maximum pooling (GMP) branches. *Red* represents the best results.

Crop	GAP	GMP	VOT2016		
			EAO	**Accuracy**	**Robustness**
	√		0.369	0.607	0.266
		√	0.367	0.601	0.308
	√	√	0.370	0.605	0.294
√	√		0.382	0.601	0.266
√		√	0.395	0.604	0.256
√	√	√	0.397	0.601	0.252

4. Conclusions

In this paper, we proposed two lightweight and efficient modules, namely the GC block and CiSK block, and integrated them into SiamRPN. The GC block can model the long-range context of template frames better, and the CiSK block gives models the ability of dynamic receptive fields. We used four large-scale datasets to train our model and used three mainstream benchmarks to evaluate the model's performance. Careful ablation study was carried out to demonstrate the positive effect of each module. Experiment results show that the proposed ACSiamRPN model has competitive performance.

Author Contributions: Conceptualization, X.Q. and Y.Z.; methodology, X.Q., H.C. and Y.Z.; software, Y.Z. and H.L.; validation, X.Z., X.Q. and H.C.; formal analysis, H.L. and H.C.; investigation, X.Q. and Y.Z.; resources, X.Q. and X.Z.; data curation, X.Q. and Y.Z.; writing–original draft preparation, X.Q. and Y.Z.; writing–review and editing, X.Q., H.C. and Y.Z.; visualization, X.Q. and Y.Z.; supervision, X.Z. and X.Q.; project administration, X.Q. and X.Z.; funding acquisition, X.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Artificial Intelligence Program of Shanghai, grant number 2019-RGZN-01077.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qin, X.; Fan, Z. Initial Matting-Guided Visual Tracking with Siamese Network. *IEEE Access* **2019**, *7*, 41669–41677. [CrossRef]
2. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
4. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 850–865.
5. Fan, H.; Ling, H. Siamese Cascaded Region Proposal Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7952–7961.
6. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware Siamese Networks for Visual Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 103–119.
7. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4591–4600.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
9. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
10. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995.
11. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2019; pp. 4282–4291.
12. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
13. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective Kernel Networks. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 510–519.
14. Qin, X.; Wu, C.; Chang, H.; Lu, H.; Zhang, X. Match Feature U-Net: Dynamic Receptive Field Networks for Biomedical Image Segmentation. *Symmetry* **2020**, *12*, 1230. [CrossRef]
15. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–19.
16. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
17. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv* **2019**, arXiv:1904.11492.
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
19. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
20. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.

21. Real, E.; Shlens, J.; Mazzocchi, S.; Pan, X.; Vanhoucke, V. YouTubeBoundingBoxes. A large high-precision human-annotated data set for object detection in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5296–5305.
22. Wu, Y.; Lim, J.; Yang, M.-H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef]
23. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Cehovin, L.; Chi, Z. The Visual Object Tracking VOT2016 Challenge Results. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherland, 8–10 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 777–823.
24. Kristan, M.; Berg, A.; Zheng, L.; Rout, L.; Van Gool, L.; Bertinetto, L.; Zhou, L. The Seventh Visual Object Tracking VOT2019 Challenge Results. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
25. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 188–203.
26. Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.V.; Tao, D. MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking. In Proceedings of the Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 749–758.
27. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Accurate Scale Estimation for Robust Visual Tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014; BMVA Press: Nottingham, UK, 2014.
28. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J.P. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef]
29. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.-M.; Hicks, S.L.; Torr, P.H.S. Struck: Structured Output Tracking with Kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2096–2109. [CrossRef]
30. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1409–1422. [CrossRef]
31. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 702–715.
32. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 472–488.
33. Nam, H.; Baek, M.; Han, B. Modeling and Propagating CNNs in a Tree Structure for Visual Tracking. *arXiv* **2016**, arXiv:1608.07242.
34. Wang, L.; Ouyang, W.; Wang, X.; Lu, H.; Lijun, W.; Wanli, O.; XiaoGang, W.; Huchuan, L. Visual Tracking with Fully Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3119–3127.
35. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
36. Zhu, G.; Porikli, F.; Li, H. Tracking Randomly Moving Objects on Edge Box Proposals. *arXiv* **2015**, arXiv:1507.08085.
37. Li, X.; Ma, C.; Wu, B.; He, Z.; Yang, M. Target-Aware Deep Tracking. In Proceedings of the Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1369–1378.
38. Lukezic, A.; Vojir, T.; Zajc, L.C.; Matas, J.; Kristan, M. Discriminative Correlation Filter with Channel and Spatial Reliability. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4847–4856.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).