

Chapter

Robust Template Update Strategy for Efficient Visual Object Tracking

*Awet Hailelassie Gebrehiwot, Jesus Bescos
and Alvaro Garcia-Martin*

Abstract

Real-time visual object tracking is an open problem in computer vision, with multiple applications in the industry, such as autonomous vehicles, human-machine interaction, intelligent cinematography, automated surveillance, and autonomous social navigation. The challenge of tracking a target of interest is critical to all of these applications. Recently, tracking algorithms that use siamese neural networks trained offline on large-scale datasets of image pairs have achieved the best performance exceeding real-time speed on multiple benchmarks. Results show that siamese approaches can be applied to enhance the tracking capabilities by learning deeper features of the object's appearance. SiamMask utilized the power of siamese networks and supervised learning approaches to solve the problem of arbitrary object tracking in real-time speed. However, its practical applications are limited due to failures encountered during testing. In order to improve the robustness of the tracker and make it applicable for the intended real-world application, two improvements have been incorporated, each addressing a different aspect of the tracking task. The first one is a data augmentation strategy to consider both motion-blur and low-resolution during training. It aims to increase the robustness of the tracker against a motion-blurred and low-resolution frames during inference. The second improvement is a target template update strategy that utilizes both the initial ground truth template and a supplementary updatable template, which considers the score of the predicted target for an efficient template update strategy by avoiding template updates during severe occlusion. All of the improvements were extensively evaluated and have achieved state-of-the-art performance in the VOT2018 and VOT2019 benchmarks. Our method (VPU-SiamM) has been submitted to the VOT-ST 2020 challenge, and it is ranked 16th out of 38 submitted tracking methods according to the Expected average overlap (EAO) metrics. VPU_SiamM Implementation can be found from the VOT2020 Trackers repository¹.

Keywords: real-time, tracking, template update, Siamese

1. Introduction

Visual object tracking (VOT), commonly referred to as target tracking, is an open problem in computer vision; this is due to a broad range of possible applications and

¹ <https://www.votchallenge.net/vot2020/trackers.html>

potential tracking challenges. Thus, it has been divided into sub-challenges according to several factors, which include: the number of targets of interest, the number of cameras, the type of data (i.e., medical, depth, thermal, or RGB images), static or moving camera, offline or online (real-time) processing.

Visual object tracking is the process of estimating and locating a target over time in a video sequence and assigning a consistent label to the tracked object across each video sequence frame. VOT algorithms have been utilized as a building block in more complex applications of computer vision such as traffic flow monitoring [1], human-machine interaction [2], medical systems [3], intelligent cinematography [4], automated surveillance [5], autonomous social navigation [6] and activity recognition [7]. Real-time visual target tracking is the process of locating and associating the target of interest in consecutive video frames while the action is taking place in real-time. Real-time visual target tracking plays an inevitable role in time-sensitive applications such as autonomous mobile robot control to keep track of the target of interest while the viewpoint is changing due to the movement of the target or the robot. In such a scenario, the tracking algorithm must be accurate and fast enough to detect sudden changes in the observed environment and act accordingly to prevent losing track of the quickly moving target of interest.

Since the start of the Visual-Object-Tracking(VOT) Real-time challenge in 2017, Siamese network-based tracking algorithms have achieved top performance and won in the VOT real-time challenge with a considerable margin over the rest of the trackers. Nearly all top ten trackers applied the siamese network, and also the winners. The dominant methodology in real-time tracking, therefore, appears to be associated. A siamese network aims to learn a similarity function. It has a Y-shaped network architecture that takes two input images and returns similarity as an output. Siamese networks are utilized to compare the similarity between the template and the candidate images to determine if the two input images have an identical pattern(similarity). In the past few years, a series of state-of-the-art siamese-based trackers have been proposed, and all of them utilize embedded features by employing CNN to compute similarity and produce various types of output, such as similarity score(probability measure), response map(two-dimensional similarity score map), and bounding box location of the target.

Luca Bertinetto et al. [8] proposed Siamese fully convolutional network (SiameseFC) to addresses the broad similarity learning between a target image and

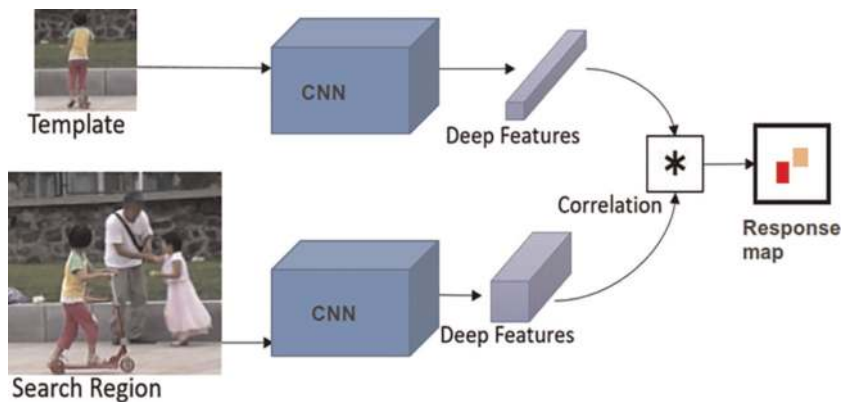


Figure 1. Fully-convolutional Siamese architecture. The output is a scalar-valued score map whose dimension depends on the size of the search image [8].

search image, as presented in **Figure 1**. According to the VOT winner rules, the winning real-time tracker of the VOT2017 [9] was SiamFC. SiamFC applies a fully-convolutional siamese network trained offline to locate an exemplar (template) image inside a larger search image **Figure 1**. The network is fully convolutional w.r.t search image: dense and efficient sliding window evaluation is achieved with a bilinear layer that computes the cross-correlation of two inputs. The deep convolutional network is trained offline with "ILSVRC VID" dataset [10] to address a general similarity learning problem and maximize target discrimination power. During tracking, SiamFC takes two images and infers a response map using the learned similarity function. The new target position is determined at the maximum value on the response map, where it depicts a maximum similarity **Figure 1**. As improvement in Siamese based tracking methods, Qiang Wang et al. [11] proposed SiamMask aiming to improve the ability of the SiamFC network to differentiate between the background and the foreground by augmenting their loss with a binary segmentation task. SiamMask is a depth-wise cross-correlation operation performed on a channel-by-channel basis, to keep the number of channels unchanged. The result of the depth-wise cross-correlation indicated as RoW (response of candidate window), then distributed into three branches, respectively segmentation, regression, and classification branches **Figure 2**.

Seven of the top ten realtime trackers (SiamMargin [12], SiamDWST [13], SiamMask [11], SiamRPNpp [14], SPM [15] and SiamCRF-RT) are based on siamese correlation combined with bounding box regression. In contrast, the top performers of the VOT2019 Real-time challenge are from the class of classical siamese correlation trackers, and siamese trackers with region proposals [16]. Although these methods showed a significant improvement, there was small attention on how to carefully update the template of the target as time goes from the start of the tracking. In all top performers, the target template is initialized in the first frame and then kept fixed during the tracking process. However, diverse variations regarding the target usually occur in the process of tracking, i.e., camera orientation, illumination change, self-rotation, self-deformation, scale, and appearance change. Thus, failing to update the target template leads to the early failure of the tracker. In such scenarios, it is crucial to adapt the target template model to the current target appearance. In addition to this, most of the tracking methods fail when motion-blurred frames or frames with low-resolution appear in the video sequence, as depicted in **Figures 3** and **4**. We

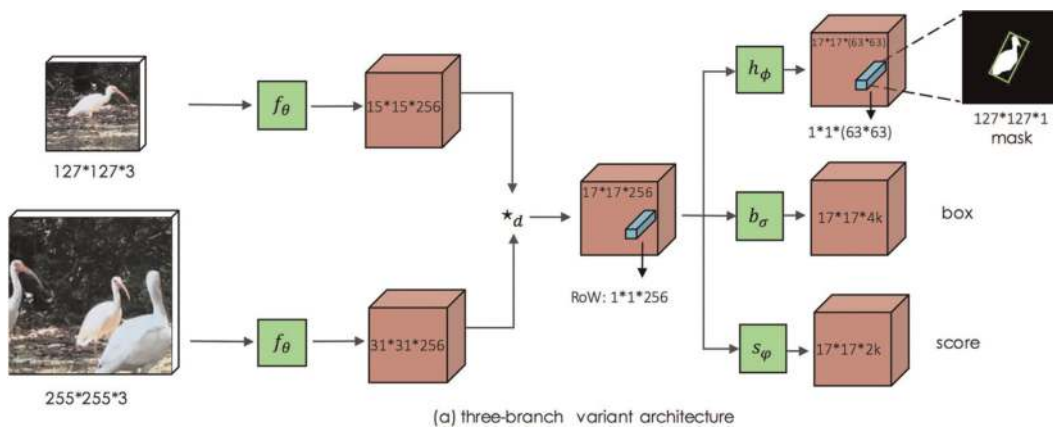


Figure 2. An illustration of SiamMask with three branches, respectively segmentation, regression, and classification branches; where $*_d$ denotes depth-wise cross correlation [11].

believe that this case arguably arises from the complete lack of similar training samples. Therefore one must incorporate a data-augmentation strategy to consider both motion-blur and low-resolution during training to significantly increase the diversity of datasets available for training without actually gathering new data.

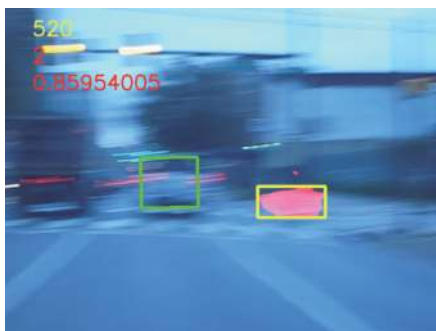
2. Method

The problem of establishing a correspondence between a single target in consecutive frames can be affected by factors such as initializing a track, updating it robustly, and ending the track. The tracking algorithm receives an input frame from the camera module and performs the visual tracking over a frame following a siamese network-based tracking approach. Since developing a new tracking algorithm from scratch is beyond the scope of this chapter, a state-of-the-art siamese-based tracking algorithm called siammask [11], one of the top performers in the VOT2019 real-time challenge, is used as a backbone of our tracking algorithm.

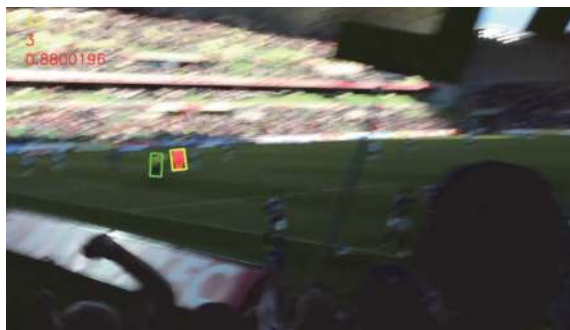
To mitigate the limitations associated with Siamese-based tracking methods. This section presents two improvements on top of the SiamMask implementation. *VPU_SiamM Implementation can be found from VOT2020 Trackers repository*².

2.1 Data-augmentation

As mentioned in the introduction, the siamese-based tracker fails when motion-blurred frames or frames with low-resolution appear in the video sequence, as depicted in **Figures 3** and **4**. Therefore to address the problems, a tracking algorithm should incorporate a data-augmentation strategy to consider both motion-blur and low-resolution during training. Since data augmentation is a strategy that significantly increases the diversity of datasets available for training without actually gathering new data, it will require implementing the data augmentation techniques explained through the following sub-sections.



(a) Tracking failure from car VOT2019 dataset



(b) Tracking failure from the soccer VOT-2019 dataset

Figure 3.

An example of SiamMask failure due to motion-blur, green and yellow bounding box indicates ground truth and predicted target respectively.

² <https://www.votchallenge.net/vot2020/trackers.html>



(a) Tracking failure from agility VOT2019 dataset



(b) Tracking failure from handball VOT-2019 dataset

Figure 4.

An example of SiamMask failure due to low resolution, green and yellow bounding box indicates ground truth and predicted target respectively.

2.1.1 Data-augmentation for motion-blur

Kernel filters are a prevalent technique in image processing to blur images. These filters work by sliding an $n \times n$ matrix across an image with a Gaussian blur filter, resulting in a blurry image. Intuitively, blurring images for data augmentation could lead to higher resistance to motion blur during testing [17]. **Figure 5** illustrates an example of a motion-blurred frame generated by the developed data-augmentation technique.

2.1.2 Data-augmentation for low-resolution

We followed a Zhangyang Wang et al. [18] approach to generate a low-resolution dataset. During training, the original (High Resolution) images are first downsampled by $scale = 4$ and then upsampled back by $scale = 4$ with nearest-neighbor interpolation as low-resolution images. A small additive Gaussian noise is added as a default data augmentation during training. An illustrates of a low-resolution frame generated by the developed data-augmentation technique is depicted in **Figure 6**.

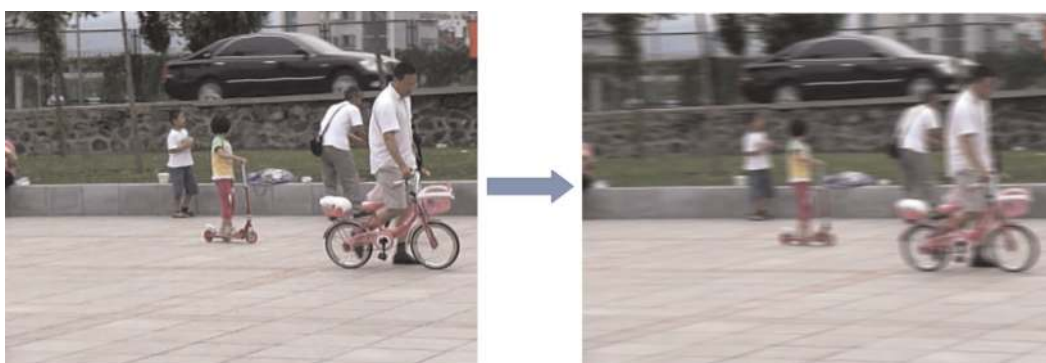
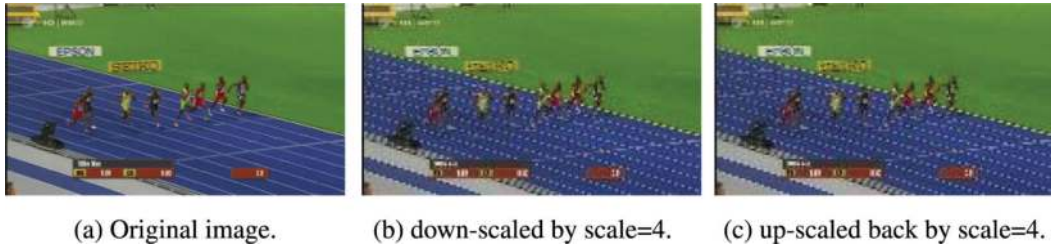


Figure 5.

An example of motion blurred frame (left image) generated from original frame (right image) using the developed data-augmentation for motion-blur technique.

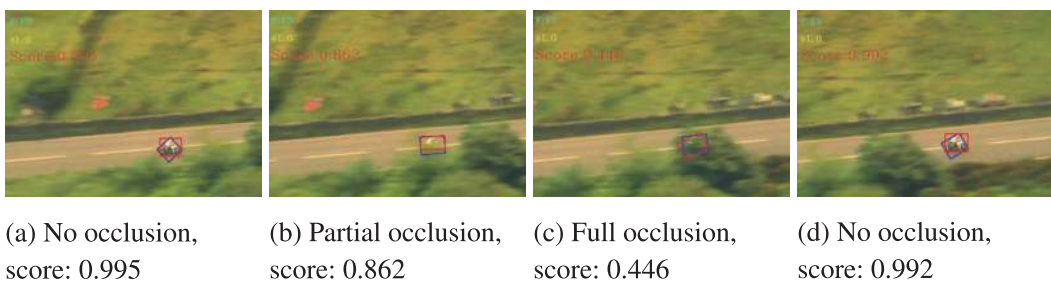
**Figure 6.**

An illustrates on how the low-resolution data augmentation generation are performed (from (a) to (c)).

2.2 Target template updating strategy

The target template update mechanism is an essential step, and its robustness has become a crucial factor influencing the quality of the tracking algorithm. To tackle this problem, more recent Siamese trackers [19–21] have implemented a simple linear update strategy using a running average with a constant learning rate. However, A simple linear update is often inadequate to cope with the changes needed and to generalize to all potentially encountered circumstances. Lichao Zhang et al. [22] proposes to replace the hand-crafted update function with a method that learns to update, using a convolutional neural network called *UpdateNet*, aims to estimate the optimal template for the next frame. However, excessive reliance on a single updated template may suffer from catastrophic drift and the inability to recover from tracking failures.

One can argue the importance of the original initial and supplementary updatable templates, which incorporate the up-to-date target information. To this end, we have incorporated a template updates strategy that utilizes both the initial template (ground truth template) T_G and an updatable template T_i . Consequently, the initial template T_G provides highly reliable information. It increases robustness against model drift, whereas an updatable template T_i integrates the new target information at the predicted target location given by the current frame. However, when a target is temporarily occluded, such as when a motorbike passes through the forest and is shielded by trees **Figure 7**, updating the template during occlusion is not required as it may cause template pollution because of shield interference. Therefore, our system needs to recognize if occlusion occurs and be able to decide whether to update the template or not. Examples of occlusion in tracking are shown in **Figures 7** and **8**. As depicted in **Figures 7** and **8**, when the target is occluded, the score becomes small, indicating the similarity between the tracked target in the current frame and the

**Figure 7.**

Overview on how the target similarity score (red) varies under different occlusion scenario during tracking process. The similarity score is indicated in red color in the top left of each frame, VOT2019 road dataset. Where blue: Ground truth, red: Tracking result.

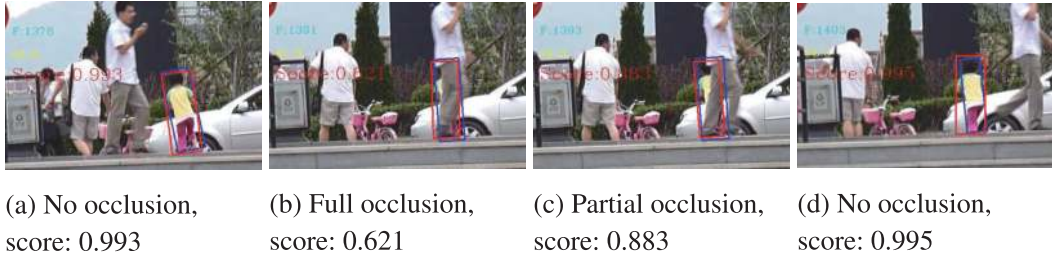


Figure 8. Overview on how the target similarity score varies under different occlusion scenario during tracking process, VOT2019 girl dataset. Where blue: Ground truth, red: Tracking result.

template is low. Thus the response on the score value can be used as the criterion for strategic decision. **Figure 9** illustrates an overview of the method.

2.2.1 Updating with a previous and a future template

In 2.2 the target template update strategy considers the target appearance only from the previous frame. However, in this section, we introduce an alternative template update strategy that considers both the target appearance from the previous frame and the target appearance in the future frame, which incorporates future information of the target appearance by updating the updatable template T_i described in 2.2. The template updating mechanism is shown in **Figure 10**. During online tracking, the template updating and the tracking procedure works as follows:

1. Tracking procedure on the next frame $i + 1$ is applied using both the previously updated target template T_i and the ground truth target template T_G .

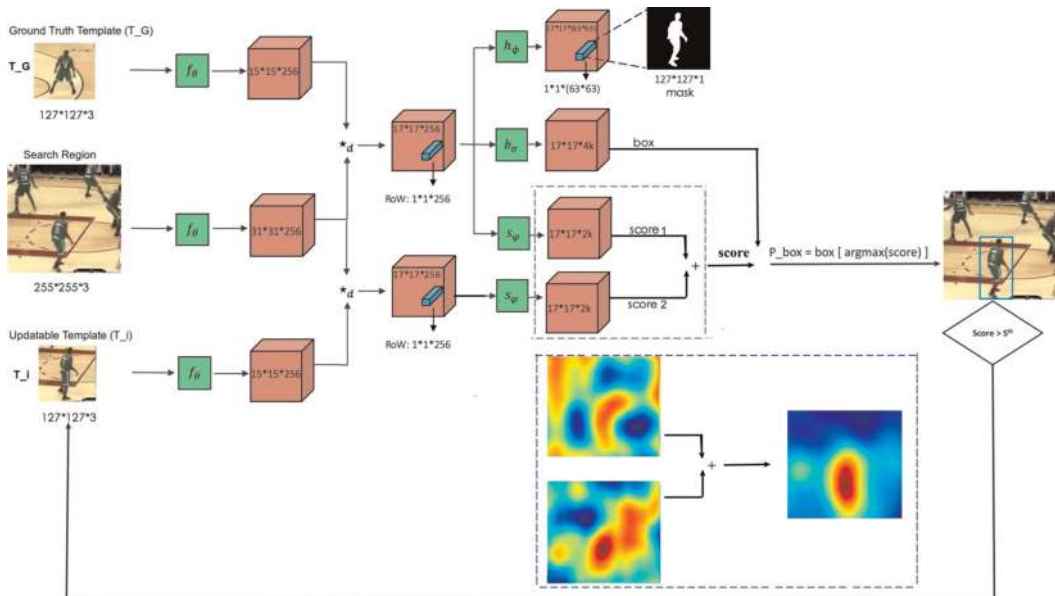


Figure 9. Target template update strategy: Where T_G is the ground truth template, T_i is an updatable template, S^{th} is the score threshold and P_box is the predicted target location.

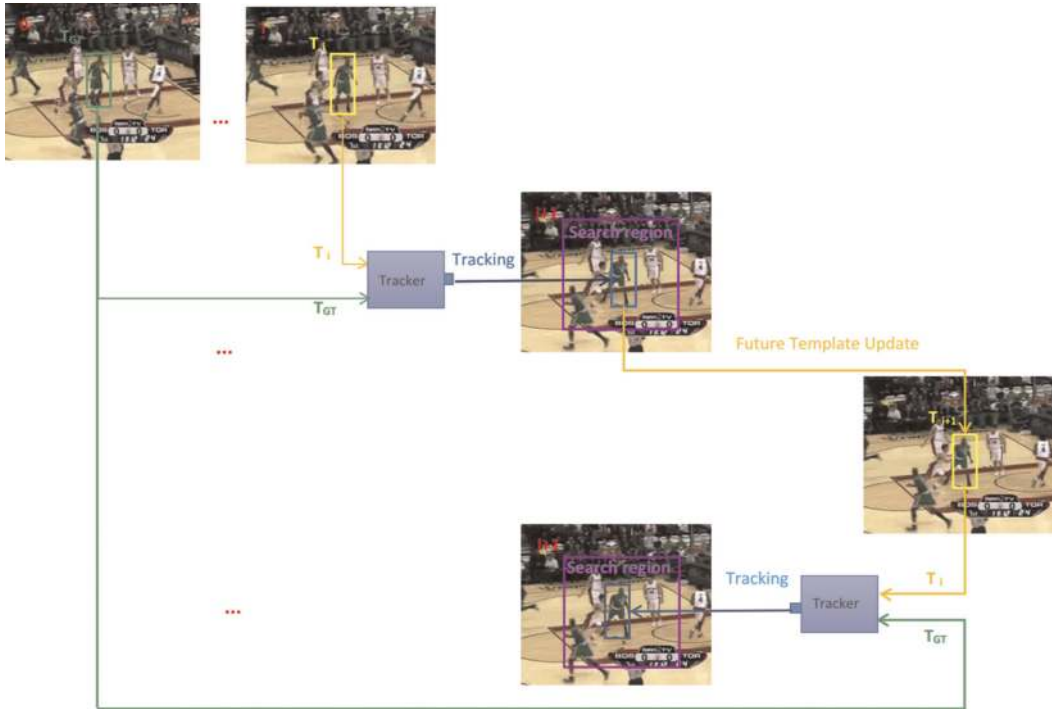


Figure 10.

Updating with a previous and a future template, where T_G is ground truth target template, T_i is previous target template and T_{i+1} is future target template. Where green: Ground truth template, yellow: Updatable template, blue: Tracking result.

2. Updatable template T_i is updated using the predicted target from the next frame to incorporate a piece of future information about the target.
3. Tracking procedure again on the current frame is applied using both the updated future target template T_{i+1} and the ground truth target template T_G .

First, a tracking procedure is applied using both the previous target template in T_i and the ground truth template T_G to perform tracking on the next frame. Then the updatable template T_i is updated using the predicted target on the next frame incorporating a piece of future information about the target. Finally, a tracking procedure is again applied to the current frame using both the updated future target template T_{i+1} and the ground truth template T_G .

3. Implementation

3.1 Training

The SiamMask implementation was trained using 4 Tesla V100 GPUs. In this experiment, only the refinement module of the mask branch is trained. The training process was carried out using COCO³ and Youtube-vos⁴ Datasets: The training was

³ <http://cocodataset.org/>

⁴ <https://youtube-vos.org/dataset/vis/>

performed over ten epochs using mini-batches of 32 samples. The data augmentation techniques described in 2.1.1 and 2.1.2 were utilized for generating datasets with motion-blur and low-resolution, respectively.

3.2 Tracking

During tracking, the tracking algorithm is evaluated once per frame. The output mask is selected from the location attaining the maximum score in the classification branch and creating an optimized bounding box. Finally, the highest scoring output of the box branch is used as a reference to crop the next search frame.

3.3 Visual-object-tracking benchmark

As object tracking has gotten significant attention in the last few decades, the number of publications on tracking-related problems has made it difficult to follow the developments in the field. One of the main reasons is that there was a lack of commonly accepted annotated datasets and standardized evaluation protocols that allowed an objective comparison of different tracking methods. To address this issue, the Visual Object Tracking (VOT) workshop was organized in association with ICCV2013⁵. Researchers from the industry and academia were invited to participate in the first VOT2013 challenge, which was aimed at model-free single-object visual trackers. In contrast to related attempts in tracker benchmarking, the dataset is labeled per-frame by visual properties such as occlusion, motion change, illumination change, scale, and camera motion, offering a more systematic comparison of the trackers [23]. VOT focused on short-term tracking (no re-detection) until the VOT2017 challenge, where a new "real-time challenge" was introduced. In the Real-time challenge, the tracker constantly receives images at real-time speed. If the tracker does not respond after the new frame becomes available, the last bounding box from the previous frame is reported as the tracking result in the current frame.

3.4 VOT evaluation metrics

The VOT challenges applies a reset-based methodology. Whenever a zero overlap between the predicted bounding box and the ground truth occurs, a failure is detected, and the tracker is re-initialized five frames after the failure. There are three primary metrics used to analyze the tracking performance in visual object tracking challenge benchmark: Accuracy (A), Robustness (R), and Expected Average Overlap (EAO) [9].

3.4.1 Accuracy

Accuracy is calculated as the average overlap between the predicted and ground truth bounding boxes during successful tracking periods [23]. The tracking accuracy at time-step t is defined as the overlap between the tracker predicted bounding box A_t^T and the ground truth bounding box A_t^G

⁵ <http://www.iccv2013.org/>

$$\Phi_t = \frac{A_t^G \cap A_t^T}{A_t^G \cup A_t^T} \quad (1)$$

3.4.2 Robustness

Robustness measures how often the tracker loses/fails the target, i.e., a zero overlap between the predicted and the ground truth bounding boxes during tracking. The protocol specifies an overlap threshold to determine tracking failure. The number of failed tracked frames are then divided by the total number of frames, as depicted in Eq. (2):

$$P_\tau = \frac{\{\Phi_t \leq \tau\}_{k=1}^N}{N} \quad (2)$$

Where τ is the overlap threshold which is zero in this case, and N is the run time of the tracker in frames. A failure is identified in a frame when the overlap (as computed using Eq. (1)) is below the defined threshold τ . Thus, the robustness of the tracker is given as a normalized number of incorrectly tracked frames.

3.4.3 Expected average overlap (EAO)

For the purpose of ranking tracking algorithms, it is better to have a single metric. Thus, in 2015 the VOT challenge introduced Expected Average Overlap (EAO), which combines both Accuracy and Robustness. EAO estimates the average overlap that a tracker is expected to achieve on a large collection of short-term sequences with the same visual properties as the given dataset.

The EAO metric can be found by calculating the average of $\hat{\Phi}_{N_s}$ over typical sequence lengths, from N_{lo} to N_{hi} :

$$\Phi = \frac{1}{N_{hi} - N_{lo}} \sum_{N_s=N_{lo}}^{N_{hi}} \hat{\Phi}_{N_s} \quad (3)$$

3.5 Experiment A: Low-resolution data-augmentation

This first experiment is dedicated to evaluating the impact of the low-resolution data-augmentation technique. The data augmentation technique described in 2.1.2 was applied to generate datasets with low-resolution during the training process of the refinement module of the network.

3.5.1 Evaluation

The performance of the developed method: incorporating low-resolution datasets using data augmentation technique during training has been evaluated using the VOT evaluation metrics on the VOT2018, VOT2019 datasets. The overall Evaluation results are shown in **Table 1**.

The term # **Tracking Failures (Lost)** indicates how often the tracking algorithm lost the target in the given video sequence, basically:

- **Tracking Lost/Failure:** is when IOU between Ground-truth and Predicted Bounding box is Zero. Thus the lower the values ↓, the higher the performance.

In **Table 1**, we compare our approach against the state-of-the-art SiamMask tracker on the VOT2018 and VOT2019 benchmarks, respectively. It can be clearly observed that the data augmentation technique for incorporating low-resolution datasets has contributed to robustness improvements. The tracker’s failure has decreased from 60 to 53 and from 104 to 93 in VOT2018 and VOT2019, respectively. Improvements are clearly shown especially in a video sequence with low-resolution, i.e. *handball1*, and *handball2* as depicted in **Figure 11**.

The results obtained in **Table 1** confirm that the developed methodology significantly improved the overall performance of the tracker. This approach outperforms the original SiamMask achieving a relative gain of 2.6% and 0.4% in EAO on VOT2018 and VOT2019, respectively. Most significantly, a gain of around 3% and 5% in Robustness value has been achieved on VOT2018 and VOT2019, respectively.

3.5.2 Results

As it is depicted in **Figure 11**, The data-augmentation for incorporating low-resolution datasets during training has contributed to enhancing the tracker robustness. Thus, the tracker becomes robust against low-resolution frames during inference in relative to the original SiamMask tracker.

VOT Metrics	VOT2018		VOT2019	
	SiamMask	Ours	SiamMask	Ours
EAO ↑	0.380	0.406	0.280	0.284
Accuracy ↑	0.609	0.589	0.610	0.586
Robustness ↓	0.279	0.248	0.522	0.467

Table 1. Comparison between SiamMask and the developed method (incorporating low-resolution data during training), under the VOT metric (EAO, Accuracy, Robustness) on VOT2018 (left) and VOT2019 (right), best results are marked in Bold.



Figure 11. Qualitative comparison between SiamMask and developed data-augmentation technique for incorporating low-resolution datasets during training. Where blue: Ground truth, red: Tracking result.

3.6 Experiment B: Motion-blur data-augmentation

In this experiment, the data-augmentation technique for incorporating motion-blurred datasets described in 2.1.1 was applied for generating datasets with motion-blur during the training process of the refinement module of the network.

3.6.1 Evaluation

The performance of the tracking algorithm incorporating the motion-blur data augmentation technique has been evaluated using the VOT evaluation metrics on the VOT2018, VOT2019 datasets. The Overall Evaluation results are shown in **Table 2**.

The data augmentation technique for incorporating motion-blurred datasets has contributed to the overall enhancement of the tracker performance. They are clear improvements in terms of Robustness in multiple video sequences relative to SiamMask. From **Table 2**, it can be concluded that the data augmentation technique for incorporating motion-blurred datasets has contributed to the improvement in Robustness of the tracker, especially in a video sequence with a motion-blur, i.e., *ball3*, and *car1*. The overall performance of the tracker has been improved, and the developed method obtained a significant relative gain of 2.1% EAO in VOT2018 and 4% R in VOT2019, compared to the SiamMask result as it is depicted in **Table 2**.

3.6.2 Results

Figure 12 presents a visual comparison between SiamMask and the developed improvement incorporating motion-blurred datasets during training using

VOT Metrics	VOT2018		VOT2019	
	SiamMask	Ours	SiamMask	Ours
EAO \uparrow	0.380	0.401	0.280	0.288
Accuracy \uparrow	0.609	0.610	0.610	0.610
Robustness \downarrow	0.279	0.248	0.522	0.482

Table 2.

Comparison between SiamMask and the developed method (incorporating motion-blurred dataset during training), under the VOT metric (EAO, accuracy, robustness) on VOT2018 (left) and VOT2019 (right).



Figure 12.

Qualitative comparison between SiamMask and developed data-augmentation technique: Incorporating motion-blurred datasets during training. Where blue: Ground truth, red: Tracking result.

data-augmentation. From **Figure 12** it can be clearly observed that the data-augmentation for incorporating motion-blurred dataset during training has contributed to enhancing the tracker Robustness. Thus, the tracker has become robust against motion-blurred video frames during inference in relative to the original SiamMask tracker.

3.7 Experiment C: Target template updating strategy

When it comes to updating the target template, the question is how and when to update the target. The parameter S^{th} controls when to update the target template according to the developed template update strategy in 2.2. Thus, the 2nd (updatable) target template is updated when the predicted target’s score is higher than the threshold S^{th} . Therefore determining the optimal threshold value S^{th} is the main focus in this sub experiment.

This set of experiments compares the effect of the target template updating strategy by varying the score threshold of S^{th} by evaluating the tracking performance on VOT2018 and VOT2019 datasets. From the experimental results shown in **Tables 3** and **4**, it can be observed that the performance of the tracker increases as the parameter S^{th} increases. Thus by using a S^{th} value as high as possible, we guaranty an efficient template update strategy by avoiding template updates during severe occlusion. **Figure 13** illustrates an overview of how each VOT metric (EAO, Accuracy, and Robustness) and FPS behave as we vary the S^{th} . Therefore, the parameter S^{th} plays an important role in deciding whether to update the target template or not when cases such as occlusion or deformation occur, as illustrated in **Figures 14** and **15**. It is worth mentioning that the template update has a negative impact on the tracker’s speed since it needs to compute the feature map of the updated template for every updated template. Therefore by setting S^{th} high, we can leverage both performance and speed as it is depicted in **Figure 13**.

Figures 14 and **15** are an illustration of how the template update strategy decides when to update the updatable template. For instance in **Figure 15a** the target is not occluded; as a result the score is high, thus *Update : True* flag is generated indicating to update the target template, on the other hand in **Figure 15b** and **c**, the target is

VOT-Metrics					
S^{th}	EAO ↑	Accuracy ↑	Robustness ↓	# Lost ↓	FPS ↑
0.65	0.377	0.602	0.267	57	25
0.7	0.371	0.602	0.267	57	27
0.75	0.385	0.600	0.248	53	28
0.8	0.387	0.603	0.258	55	31
0.85	0.388	0.603	0.243	52	32
0.9	0.393	0.602	0.239	51	35
0.95	0.397	0.602	0.239	51	40

Table 3. Determining the optimal score threshold (S^{th}) for updating the target template under VOT-metrics on VOT2018.

VOT-metrics					
S^{th}	EAO \uparrow	Accuracy \uparrow	Robustness \downarrow	# Lost \downarrow	FPS \uparrow
0.65	0.276	0.598	0.497	99	25
0.7	0.278	0.601	0.497	99	26
0.75	0.278	0.601	0.497	99	27
0.8	0.278	0.601	0.497	99	27
0.85	0.274	0.601	0.512	102	32
0.9	0.278	0.600	0.512	102	36
0.95	0.293	0.601	0.482	96	41

Table 4. Determining the optimal score threshold (S^{th}) for updating the target template under VOT-metrics on VOT2019.

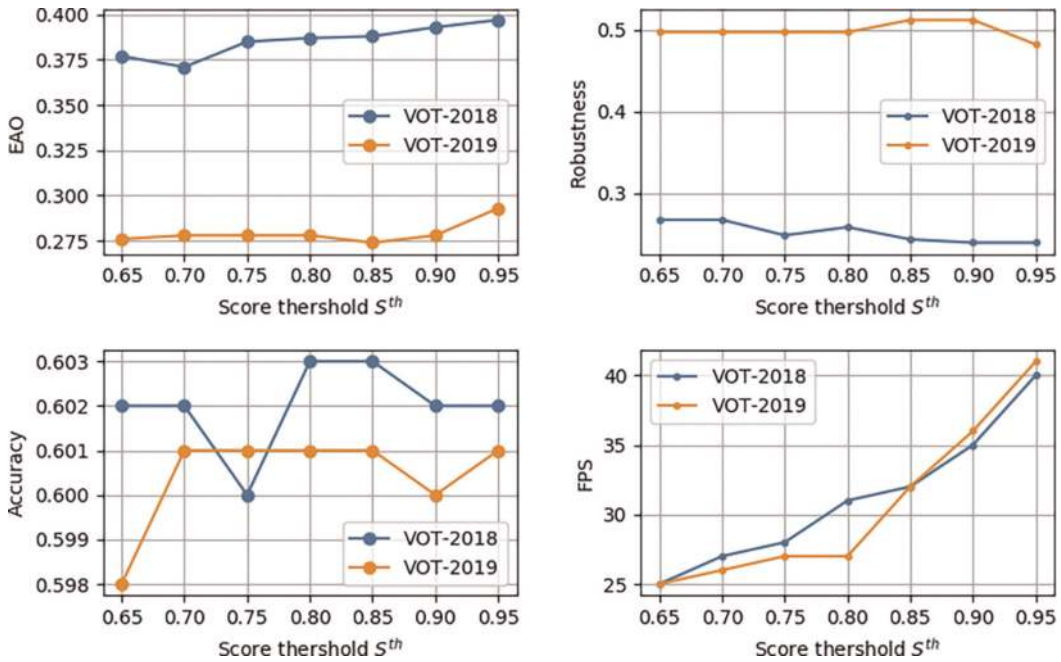


Figure 13. An illustration on the effect of the tracking performance, with a template update strategy by varying the score threshold S^{th} , NB: This experiments carried out using the checkpoint which include data-augmentation.

occluded by the tree: thus *Update* : *False* flag is generated indicating not to update the target template during such occlusions. This experiment was carried out using $S^{th} > 0.95$.

Table 5 presents a comparison between no-update SiamMask and incorporating the developed template update strategy: it can be observed that a relative gain of 0.7% and 2.0% in Robustness has been achieved by incorporating template update strategy. Thus, the tracker has encountered less failure than the no-update SiamMask, decreasing from 60 to 58 and 104 to 100 in VOT2018 and VOT2019 benchmarks, respectively. The robustness of the tracker is the crucial element for applications such as automatic robotic cameras where there is no human assistance.



Figure 14. Visual illustration on how the target template update strategy decides whether to update the template or not based on the similarity score under different occlusion scenario during tracking process, VOT2019 girl dataset. Where blue: Ground truth, red: Tracking result.

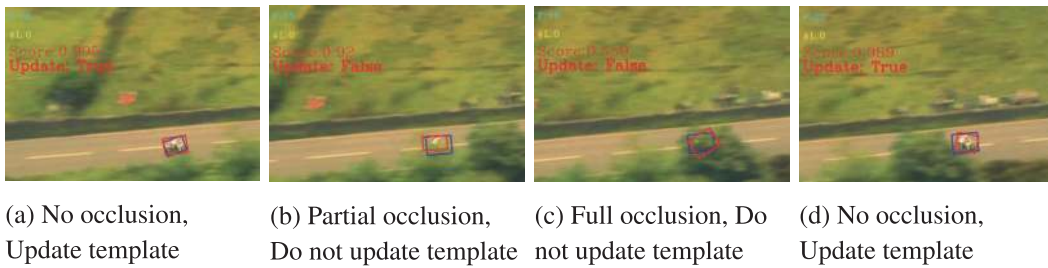


Figure 15. Visual illustration on how the target template update strategy decides whether to update the template or not based on the similarity score under different occlusion scenario during tracking process, VOT2019 girl dataset. Where blue: Ground truth, red: Tracking result.

	VOT2018		VOT2019	
	SiamMask	Ours	SiamMask	Ours
EAO ↑	0.380	0.351	0.280	0.268
Accuracy ↑	0.609	0.593	0.610	0.593
Robustness ↓	0.279	0.272	0.522	0.502
# Lost ↓	60	58.0	104	100.0
FPS ↑	44	40	44	40

Table 5. Comparison between no-update SiamMask and incorporating target template update under VOT2018 (left) and VOT2019 (right) benchmarks.

3.7.1 Updating with a previous and a future template

This experiment dedicated to examine the strength and weakness of the ”updating with a previous and future frame” template update strategy described in 2.2.1. As can be seen from **Table 6**, the method ”updating with previous and a future template” has achieved a relative gain of around 0.7% and 2.5% in Robustness value w.r.t SiamMask in both VOT2018 and VOT2019 benchmark, respectively. This indicates that the ”Updating with Previous and Future template” strategy has enhanced the tracker’s Robustness, which is the most crucial in automated tracking applications. However, this can not be used for real-time applications as the processing speed is very slow, around 12 FPS on a laptop equipped with NVIDIA GEFORCE GTX1060. The main

	VOT2018		VOT2019	
	SiamMask	Ours	SiamMask	Ours
EAO \uparrow	0.380	0.357	0.280	0.274
Accuracy \uparrow	0.609	0.597	0.610	0.597
Robustness \downarrow	0.279	0.272	0.522	0.497
# Lost \downarrow	60	58	104	99
FPS \uparrow	44	12	44	12

Table 6.

Comparison between no-update SiamMask and incorporating target template updating with previous and future template on VOT2018 (left) and VOT2019 (right) benchmark.

computational burden on the tracker is related to the target template feature extraction network. Thus, the tracking algorithm processing speed becomes very slow when the target template is updated with the previous and future template, resulting in a poor FPS.

3.8 Experiment E: Comparison with state-of-the-art trackers

This section compares our tracking framework called VPU_SiamM with other state-of-the-art trackers SiamRPN, SiamMask in the VOT2018 and SiamRPN++, SiamMask in VOT2019.

To take advantage of the incorporated improvements, a tracker named VPU_SiamM has been developed. VPU_SiamM has been trained based on the data augmentation technique incorporating both motion-blur and low-resolution, and during online inference, a target template update strategy is applied.

We have tested our VPU_SiamM tracker on the VOT2018 dataset in comparison with state-of-the-art methods. We compare with the top trackers SiamRPN (winner of the VOT2018 real-time challenge) and SiamMask among the top performer in the VOT2019 challenge. Our tracker obtained a significant relative gain of 1.3% in EAO, compared to the top-ranked trackers. Following the evaluation protocol of VOT2018, we adopt the Expected Average Overlap (EAO), Accuracy (A), and Robustness (R) to compare different trackers. The detailed comparisons are reported in **Table 7**: it can be observed that the VPU_SiamM has achieved top performance on EAO, and R. Especially, our VPU_SiamM tracker outperforms SiamRPN (the VOT2018 real-time challenge winner), achieving a relative gain of 1.3% in EAO and 1.6% in Accuracy and 4% in Robustness. Besides, our tracker yields a relative gain of 4% on Robustness w.r.t

Tracker	VOT2018		
	EAO \uparrow	Accuracy \uparrow	Robustness \downarrow
SiamRPN [21]	0.383	0.586	0.276
SiamMask [11]	0.38	0.609	0.279
VPU_SiamM	0.393	0.602	0.239

Table 7.

Comparison of our tracker VPU_SiamM with the state-of-the-art trackers SiamRPN and SiamMask in terms of expected average overlap (EAO), accuracy, and robustness (failure rate) on the VOT2018 benchmark.

VOT2019			
Tracker	EAO \uparrow	Accuracy \uparrow	Robustness \downarrow
SiamRPN++ [14]	0.282	0.598	0.482
SiamMask [11]	0.287	0.594	0.461
VPU_SiamM	0.293	0.601	0.482

Table 8.

Comparison of our tracker VPU_SiamM with the state-of-the-art trackers SiamRPN++ and SiamMask in terms of expected average overlap (EAO), accuracy, and robustness (failure rate) on the VOT2019 benchmark.

both SiamMak and SiamRPN, which is the common vulnerability of the Siamese network-based trackers.

Following previous VOT evaluation, we have evaluated our VPU_SiamM tracker on VOT2019 datasets, which contains 60 challenging testing sequences. As shown in **Table 8**, our VPU_SiamM also achieves the best tracking results on VOT2019 in EAO and Accuracy metrics compared to state-of-the-art trackers SiamMask and SiamRPN+. More specifically, our approach improves the EAO by around 1%.

Submission to VOT-ST 2020 Challenge: Our method (VPU-SiamM) has been submitted to the VOT-ST 2020 challenge [24], and our tracking methods (VPU SiamM) is ranked 16th out of 38 computing tracking methods according to the Expected average overlap (EAO) metrics [24].

4. Conclusions

In this chapter, one of the state-of-the-art tracking algorithms based on siamese networks called SiamMask has been used as a backbone, and two improvements have been affixed, each addressing different aspects of the tracking task.

The developed data augmentation technique for incorporating low-resolution and motion-blur has been evaluated separately and jointly, achieving state-of-the-art results in the VOT2018 and VOT2019 benchmarks. From the evaluation results, it is clear to conclude that the data augmentation technique has played an essential role in improving the overall performance of the tracking algorithm. It has outperformed the SiamMask results in both VOT2018 and VOT2019 benchmarks. In contrast, among the three data augmentation techniques, the data augmentation technique for incorporating both motion-blur and low-resolution outperforms the rest in terms of EAO in VOT2018 and VOT2019 benchmarks. Nevertheless, the data-augmentation for incorporating only motion-blur has achieved a top performance according to the Accuracy metric in both VOT2018 and VOT2019 benchmarks. However, the Accuracy is less significant as it only considers the IOU during a successful tracking. According to the VOT ranking method, the EAO value is used to rank tracking methods. Therefore the data augmentation technique for incorporating both motion-blur and low-resolution is ranked top among the others. This indicates that the data-augmentation technique has contributed to the improvement of the overall tracker performance.

Comparable results on VOT2018 and VOT2019 benchmarks confirm that the robust target template update strategy that utilizes both the initial ground truth template and a supplementary updatable template and avoiding template updates during severe occlusion can significantly improve the tracker's performance with respect to SiamMask results while running at 41 FPS.

A tracker named VPU_SiamM was trained based on the presented approach, and it was ranked 16th out of 38 submitted tracking methods in the VOT-ST 2020 challenge [24].

Acknowledgements

This work has been partially supported by the Spanish Government through its TEC2017-88169-R MobiNetVideo project.

Author details


Awet Hailelassie Gebrehiwot^{1*}, Jesus Bescos² and Alvaro Garcia-Martin²

1 Czech Technical University in Prague, Prague, Czech Republic

2 Universidad Autonoma de Madrid, Madrid, Spain

*Address all correspondence to: awethailelassie21@gmail.com

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Tian B, Yao Q, Gu Y, Wang K, Li Y. Video processing techniques for traffic flow monitoring: A survey. In: 14th International IEEE Conference on Intelligent Transportation Systems, ITSC 2011, Washington, DC, USA, October 5-7, 2011. IEEE; 2011. pp. 1103-1108
- [2] Zeng M, Guo G, Tang Q. Vehicle human-machine interaction interface evaluation method based on eye movement and finger tracking technology. In: HCI International 2019 - Late Breaking Papers - 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26-31, 2019, Proceedings (C. Stephanidis, ed.), vol. 11786 of Lecture Notes in Computer Science. Springer; 2019. pp. 101-115
- [3] Brandes S, Mokhtari Z, Essig F, Hünninger K, Kurzai O, Figge MT. Automated segmentation and tracking of non-rigid objects in time-lapse microscopy videos of polymorphonuclear neutrophils. *Medical Image Anal.* 2015;**20**(1):34-51
- [4] Nägeli T, Alonso-Mora J, Domahidi A, Rus D, Hilliges O. Real-time motion planning for aerial videography with real-time with dynamic obstacle avoidance and viewpoint optimization. *IEEE Robotics Autom. Lett.* 2017;**2**(3):1696-1703
- [5] Esterle L, Lewis PR, McBride R, Yao X. The future of camera networks: Staying smart in a chaotic world. In: Arias-Estrada MO, Micheloni C, Aghajan HK, Camps OI, Brea VM, editors. Proceedings of the 11th International Conference on Distributed Smart Cameras, Stanford, CA, USA, September 5-7, 2017. ACM; 2017. pp. 163-168
- [6] Chen YF, Everett M, Liu M, How JP. Socially aware motion planning with deep reinforcement learning. *CoRR.* 2017;**abs/1703.08862**
- [7] Aggarwal JK, Xia L. Human activity recognition from 3d data: A review. *Pattern Recognition Letters.* 2014;**48**: 70-80
- [8] Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PHS. Fully-convolutional siamese networks for object tracking. In: Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II (G. Hua and H. Jégou, eds.), vol. 9914 of Lecture Notes in Computer Science. 2016. pp. 850-865
- [9] Matej Kristan EA, Matas J. The visual object tracking VOT2017 challenge results. In: 2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017. IEEE Computer Society; 2017. pp. 1949-1972
- [10] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision.* 2015;**115**(3):211-252
- [11] Wang Q, Zhang L, Bertinetto L, Hu W, Torr PHS. Fast online object tracking and segmentation: A unifying approach. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE; 2019. pp. 1328-1338
- [12] Zhou J, Wang P and Sun H. Discriminative and Robust Online Learning for Siamese Visual Tracking. 2019
- [13] Zhang Z, Peng H, Wang Q. Deeper and wider siamese networks for real-

time visual tracking. CoRR. 2019;**abs/1901.01660**

[14] Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: arXiv preprint arXiv: 1812.11703. 2018

[15] Wang G, Luo C, Xiong Z, Zeng W. Spm-tracker: Series-parallel matching for real-time visual object tracking. CoRR. 2019;**abs/1904.04452**

[16] Matej Kristan EA. The seventh visual object tracking VOT2019 challenge results. In: 2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019. IEEE; 2019. pp. 2206-2241

[17] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. *J. Big Data*. 2019;**6:60**

[18] Wang Z, Chang S, Yang Y, Liu D, Huang TS. Studying very low resolution recognition using deep networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society; 2016. pp. 4792-4800

[19] Wang Q, Teng Z, Xing J, Gao J, Hu W, Maybank SJ. Learning attentions: Residual attentional siamese network for high performance online visual tracking. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society; 2018. pp. 4854-4863

[20] Zhu Z, Wang Q, Li B, Wu W, Yan J, Hu W. Distractor-aware siamese networks for visual object tracking. In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich,*

Germany, September 8-14, 2018, Proceedings, Part IX (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), vol. 11213 of *Lecture Notes in Computer Science*. Springer; 2018. pp. 103-119

[21] Li B, Yan J, Wu W, Zhu Z, Hu X. High performance visual tracking with siamese region proposal network. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society; 2018. pp. 8971-8980

[22] Zhang L, Gonzalez-Garcia A, van de Weijer J, Danelljan M, Khan FS. Learning the model update for siamese trackers. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. IEEE; 2019. pp. 4009-4018

[23] Kristan EAM. The visual object tracking vot2013 challenge results. In: 2013 IEEE International Conference on Computer Vision Workshops. 2013. pp. 98-111

[24] Kristan M, Leonardis A, Matas J, Felsberg M, Pflugfelder R, Kämäräinen J-K, et al. "The eighth visual object tracking vot2020 challenge results," in *Computer Vision – ECCV 2020 Workshops* (A. Bartoli and A. Fusiello, eds.), (Cham), Springer: International Publishing; 2020. pp. 547-601