Understanding Siamese Networks for Visual Object Tracking

by

Wen Xu

A report submitted in partial fulfillment of
the requirements for the degree of

Master of Science

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2020

The work is under supervision of:
Yu Hen Hu, Professor, Electrical and Computer Engineering

Understanding Siamese Networks for Visual Object Tracking

Wen Xu
Advisor: Yu Hen Hu
Department of Electrical and Computer Engineering
University of Wisconsin–Madison

Abstract

Visual tracking of an arbitrary object has drawn great attention in the computer vision community for years. Visual object tracking methods based on Siamese networks achieved state-of-the-art results with balanced accuracy and speed in the past few years. Understanding those trackers may shed light on future research on this topic, which makes it no less important than just incrementally improving the performance of the trackers. In this report, we give a comprehensive analysis of multiple representative trackers based on Siamese Networks. Specifically, we depict the developments of Siamese networks in visual object tracking from the earliest one Siamese-FC to the latest ones. Besides, we give both quantitative and qualitative analysis of the performance and discrimination power of Siamese networks based trackers, by evaluating them on several canonical tracking benchmarks including VOT and OTB datasets and visualizing feature maps. We also identify several causes of failures. Some future research directions are proposed in the end.

1. Introduction

Visual object tracking has been a fundamental and challenging task in the field of computer vision over the last decades. It is usually defined as firstly determining a generic object solely by the annotation, often a bounding box, of the object in the first frame of a video sequence and then tracking the object in all subsequent frames through automatically outputting the states of the object in those frames [36]. It has a significant number of applications in various fields, such as surveillance systems [39], visual analysis [9], automatic driving [34] and human-computer interactions [26].

The majority of modern tracking algorithms, i.e., trackers, can be roughly divided into two approaches. One approach is based on correlation filters. Correlation filters based trackers perform the training and tracking in an online fashion. They model the appearance of objects using filters trained on example images and compute the correlation in the Fourier domain [4, 13, 8]. Another approach is based on deep learning. These types of trackers are often trained offline and perform online tracking with or without fine-tuning. Some representative trackers are GOTURN [12], Siamese-FC [3], MDNet [28] and ATOM [7]. Among the deep learning approaches, the Siamese networks based trackers have achieved the state-of-the-art results in the VOT challenges [20, 17, 16, 18, 19] and the OTB datasets [41, 40]. Fully-Convolutional Siamese Networks (Siamese-FC) [3] is a landmark tracker that successfully applies Siamese networks to visual object tracking. Based on this Siamese networks structure, numerous trackers have been proposed [24, 43, 23, 42, 38]. The Siamese networks in those trackers are trained end-to-end offline to learn a feature extraction function which can be further used as a similarity metric between the template and the search image. At the inference time, the tracking is often formulated as a one-shot detection task with only one fixed exemplar image for each video sequence.

In this report, we will investigate the evolution of Siamese networks based trackers, analyze the results of the experiments of various Siamese networks based trackers on canonical benchmarks, and propose some future research directions.

The rest of this report is organized as follows. Section 2 summarizes related work in Siamese networks, trackers based on Siamese networks, and the one-shot learning paradigm; Section 3 gives a very detailed description on the evolution of Siamese networks and trackers based on Siamese networks structures; Section 4 evaluates multiple trackers on several canonical tracking benchmarks and shows the quantitative and qualitative analysis of the results. Section 5 concludes the report and points out some future research directions.

2. Related Work

We briefly review the related work in three aspects: Siamese networks, Siamese networks based visual object tracking, and one-shot learning.

2.1. Siamese networks

A Siamese neural network, which is also called a twin neural network, is a neural network that takes two input vectors and applied the same weights and architecture on them to compute comparable output vectors. Based on the two output feature vectors, the specific task can be done directly using the similarity or the distance between them or feed the two outputs into some further neural networks, depending on the specific task. An earlier application of Siamese neural networks is signature verification [14]. Two identical sub-networks extract features from two signatures, where one is the template, another one is the signature needed to be verified. Verification is as simple as comparing the distance between the two feature vectors and a chosen threshold. Since the template of a signer is a fixed signature, in the actual system, the feature vector of each signer is stored and in the inference time, only the candidate signatures go through one branch of the Siamese network and the candidate signature is chosen if it is closest to the stored representation. Other applications include face verification [6, 1] and image recognition [15].

2.2. Siamese networks based visual object tracking

Fully-Convolutional Siamese Networks (Siamese-FC) is one of the very first trackers based on Siamese networks to achieve both high accuracy and speed [3]. It takes advantage of the fully-convolutional property of the architecture on the search image to facilitate evaluations on all subwindows of the search image. GOTURN [12] also employs the Siamese architecture but it is trained to regress directly from the image pairs to predict a rectangle instead of a position and does not possess the transition invariance property of the second image. Following the idea of Siamese-FC, Siamese-RPN is proposed to use Siamese networks for feature extraction and an extra region proposal network for proposal refinement [24]. The time-consuming multi-scale test is no longer needed in Siamese-RPN, which significantly boosts the speed of the tracker. To learn a more discriminative feature representation, the distractor-aware Siamese networks (DaSiamRPN) [43] introduce an effective sampling strategy during training and a distractor-aware module is embedded during inference. All aforementioned trackers use similar 5 convolutional-layer AlexNet-like architecture as the backbone architecture of Siamese networks. SiamRPN++ [23] goes beyond this by developing a spatial aware sampling strategy and enables as backbones modern deep neural networks like ResNet [11] to achieve better performance using a fusion of feature maps. SiamMask [38] proposes a unifying approach for performing both object tracking and segmentation by adding a new branch for segmentation. It also enables rotated bounding box prediction for the tracking task based on the segmentation result.

Some Siamese network based trackers also integrate correlation filters. CFNet [37] overcomes the limitation of correlation filters, e.g., manually designed features and features trained from other tasks instead of tracking, by interpreting the correlation filter learner as a differentiable layer in a deep neural network. DCFNet [29] presents a more compact and much faster end-to-end lightweight network that also treats Discriminant Correlation Filters (DCF) as a special correlation filter layer added in a Siamese network, and preserve the efficiency property of DCF.

2.3. One-shot learning

One-shot learning is the task of learning information about object categories from one or a few training samples instead of hundreds or more samples. In [27], a data-defined model is developed by minimizing the summed pixel-wise entropy over a continuous set of transforms on images. The probability density over the set of transforms can be used to develop the classifier. In [25], a new Bayesian approach is proposed, where object categories are represented by probabilistic models and prior knowledge of categories is represented as a probability density function. Another set of approaches falls into the "learning to learn" paradigm. In [2], the learner, named Learnnet, itself is modeled as a second deep network, which can predict the parameters of a pupil network from just one exemplar.

3. Evolution of Siamese visual tracking

In this section, we discuss the evolution of visual object tracking based on Siamese networks in great detail. The discussions are two-fold: how visual object tracking methods take advantage of Siamese networks, which is originally proposed for verification tasks, and how those Siamese networks based trackers have developed in recent years. We not only describe the architecture, training, and testing of the models but also explain the intrinsic connections between them.

3.1. Towards Siamese networks for visual object tracking

Siamese networks. Siamese networks are an end-to-end metric learning approach. The architecture of a typical Siamese network is shown in Fig. 1. A pair of inputs z and x, usually images, is fed into two identical Siamese subnetworks. The Siamese networks will produce comparable output vectors $\varphi(z)$ and $\varphi(x)$ respectively for the two inputs, which can be viewed as the encoding/embedding of the corresponding inputs. Whether the pair of images are from the same class (of the same object, of faces from the same person) can be further determined based on the distance or the similarity score of the two vectors, which is represented by $g(\varphi(z), \varphi(x))$ in the figure, and some well-chosen thresholds.

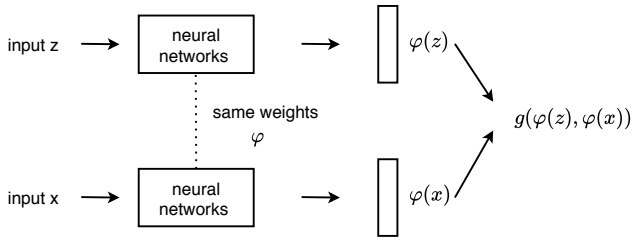


Figure 1: The architecture of Siamese networks.

Training Siamese networks. Siamese networks can be trained in multiple ways. One way is to use the triplet loss if we have multiple examples in each class. Denote the template/anchor image of a class as A, denote a positive example, i.e., an example of the same class as A, as P, and denote a negative example, i.e., an example of another class other than A, as N. Let g be the Euclidean distance function. We call the pair of A and P a positive pair as both of them are from the same class and the pair of A and N a negative pair because they are of different classes. The triplet loss function can be defined as:

$$l(A, P, N) = \max(g(\varphi(A), \varphi(P)) + \alpha - g(\varphi(A), \varphi(N)), 0), \tag{1}$$

where hyperparameter $\alpha > 0$ is the margin that is the desired difference between any positive pairs and negative pairs. The total loss L is the sum of losses of all training triplets. Suppose there are M triplets in the training set, we have

$$L = \sum_{i=1}^{M} l(A_i, P_i, N_i). \tag{2}$$

To understand the triplet loss in Eqn. 1, we need to investigate when the triplet (A, P, N) incurs the triplet loss. Clearly, the triplet (A, P, N) incurs loss of value $g(\varphi(A), \varphi(P)) + \alpha - g(\varphi(A), \varphi(N))$ if $g(\varphi(A), \varphi(P)) + \alpha \geq g(\varphi(A), \varphi(N))$. This means if the distance of a positive pair plus the margin is no less than the distance of a negative pair, the loss will be incurred. This also matches with the intuition that since φ computes the embedding of the inputs, the distance of two inputs of the same class should be close in the Euclidean space while the distance of two inputs of different classes should be large. When $g(\varphi(A), \varphi(P))$ is less than $g(\varphi(A), \varphi(N))$ minus the margin, no loss will incur for the triplet (A, P, N). The positive margin α is required here to ensure that the neural networks do not learn the trivial solution where all values of $g(\varphi(A), \varphi(P))$ and $g(\varphi(A), \varphi(N))$ are zeros and therefore achieve zero total loss.

Another way of training Siamese networks is to add a logistic regression module after the embedding module and make the problem a binary classification problem in order to use the logistic loss. This approach is especially suitable if there is no more than one template/anchor image of each class. Let y be the final binary classification result. Suppose $y = 1$ means the input pairs are of the same class and $y = 0$ means the input pairs are of different classes. Denote the length of the embedding vector $\varphi(\cdot)$ as L. Denote the weights vector $w \in \mathbb{R}^L$ and bias $b \in \mathbb{R}$. Define $s \in \mathbb{R}^L$ where $s_i = |\varphi(z)_i - \varphi(x)_i|$ for $i \in [L]$. The predicted $\hat{y}$ can be determined as:

$$\hat{y} = \begin{cases} +1 & \text{if } w^T s + b \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Hence, the logistic loss of the input pair (z, x) with ground truth label y can be defined as:

$$l(z, x, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}). \tag{4}$$

The total loss is the summation of the logistic loss of every pair in the training set.

Siamese-FC. Fully-Convolutional Siamese Networks (Siamese-FC) use the Siamese networks as the feature extractors and compare the similarities of different positions in the search image to determine the position of the object in each frame [3]. It also utilizes the property called the fully-convolutional property that makes searching feasible and is adopted in most of the subsequent Siamese networks based trackers.

The architecture of Siamese-FC is shown in Fig. 2, which is very similar to the generic Siamese networks. The input z of size $127 \times 127 \times 3$ is the exemplar image which is derived from the ground truth bounding box of the object in the first frame. The input x of size $255 \times 255 \times 3$ is the search image of each frame. By going through the same sub-networks φ of the same weights, the input x will be encoded as a feature vector of size $6 \times 6 \times 128$ and the input z will be embedded as a feature vector of size $22 \times 22 \times 128$. Cross-correlation is performed on the two feature maps to get a score of size $17 \times 17 \times 1$. The position of the highest score in the score map is reverted back to its corresponding position in x, which is chosen as the predicted bounding box center of that frame.

The fully-convolutional property. One key difference between the Siamese-FC and the plain Siamese networks is that the sizes of the two inputs in Siamese-FC are different. The authors argue that by utilizing the fully-convolutional property, the Siamese-FC is able to evaluate multiple positions of the search images at one time instead of feeding each patch of the same size as z in x to get a result for each position separately [3]. A function is fully-convolutional if it commutes with translation. In other words, for a signal, the result of applying a translation first then applying

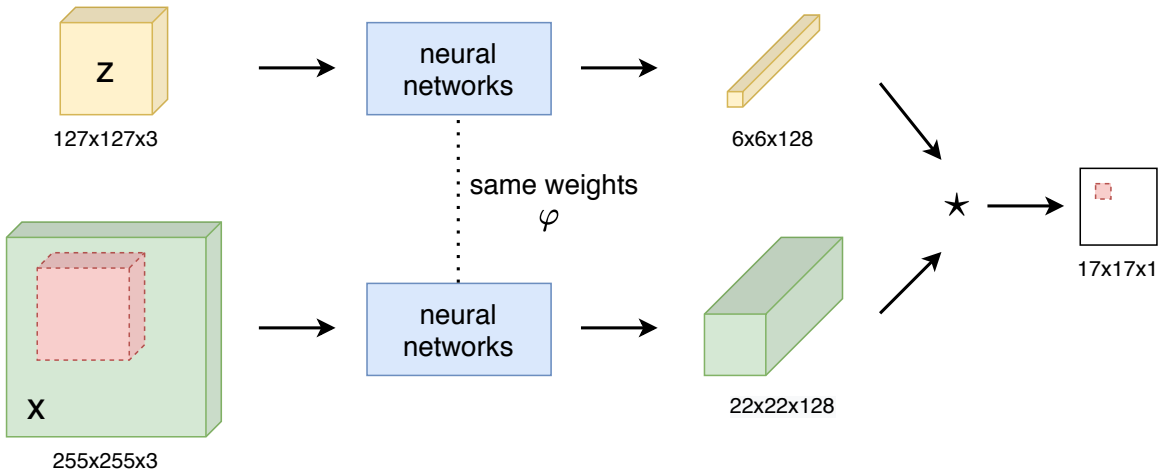


Figure 2: The architecture of Siamese-FC.

the function to the translated signal is the same as the result of applying the function to the signal first then applying the translation, conditioned on the fact that the function is fully-convolutional. If the signal is finite, this only needs to be true for the valid region of the output. Mathematically, let x be a signal, L is the translation operator, i.e., $(L_\tau x)[u] = x[u - \tau]$, and a function φ that maps signals to signals. If φ satisfies:

$$\varphi(L_{k\tau} x) = L_\tau \varphi(x), \tag{5}$$

we say the function φ is fully-convolutional with stride k. For neural networks without any zero-padding, they are fully-convolutional functions with different strides k. Siamese-FC uses AlexNet-like neural networks which have a stride of 8 [22]. The fully-convolutional property enables a search image to be used in each frame that is much larger than the fixed exemplar image and calculation of a dense 2D score map for all translated sub-windows in a single evaluation. Specifically, in Siamese-FC, this can be represented as:

$$g(z, x) = \varphi(z) \star \varphi(x) + b\mathbb{1}, \tag{6}$$

where the function g calculates the 2D score map from the input pair (z, x) based on the cross-correlation between $\varphi(z)$ and $\varphi(x)$ and the bias term $b\mathbb{1}$. The score map of Siamese-FC is of size 17×17 which is very coarse-grained compared to the input size: one-pixel displacement in the score map is eight-pixel displacement in the search image. Upsampling is performed on the score map of original size 17×17 to the size of 272×272 using bicubic interpolation, in order to get a more accurate displacement in the search image. Besides, since the position change of the object between two consecutive frames may not be large, a cosine window is applied to the upsampled score map to penalize large displacements, decreasing the chances of incorrect tracking of distractors.

Dataset curation. However, one may notice that the ground truth objects in video sequences may have sizes of large variances while the exemplar image fed into the networks must be of fixed size $127 \times 127 \times 3$ and the search image fed into the networks must be of fixed size $255 \times 255 \times 3$ based on the architecture of Siamese-FC. For the exemplar image, the authors scale the ground truth bounding boxes of the objects with some context to fit the size into the required size of z. Suppose the ground truth bounding box of an object is of size (w, h), where w is the width and h is the height. The scale factor s is chosen based on the formula:

$$s(w + p) \times s(h + p) = A, \tag{7}$$

where $p = (w + h)/2$ is the context and A is the area of exemplar image, i.e., 127^2. For the search image of each frame, it is centered at the predicted center of the previous frame and is properly scaled to fit the size to 255×255. By scaling the images, objects of arbitrary sizes can be tracked using Siamese-FC. Nonetheless, the scale change of the object itself between frames is not able to be captured in Siamese-FC. To overcome this issue, multiple scales, usually 5 or 3, are used in every frame in the inference time by assembling a mini-batch of scaled images. Hence, the final position and scale of the predicted bounding box in each frame are determined by the position of the max score in the score map of size $17 \times 17 \times n$ or $272 \times 272 \times n$, where n is number of scales. Besides, penalization is applied to any change in scale and updates of the current scale are damped.

Training Siamese-FC. Siamses-FC is trained similar to the second approach mentioned in this subsection. The training set is a set of positive and negative pairs derived

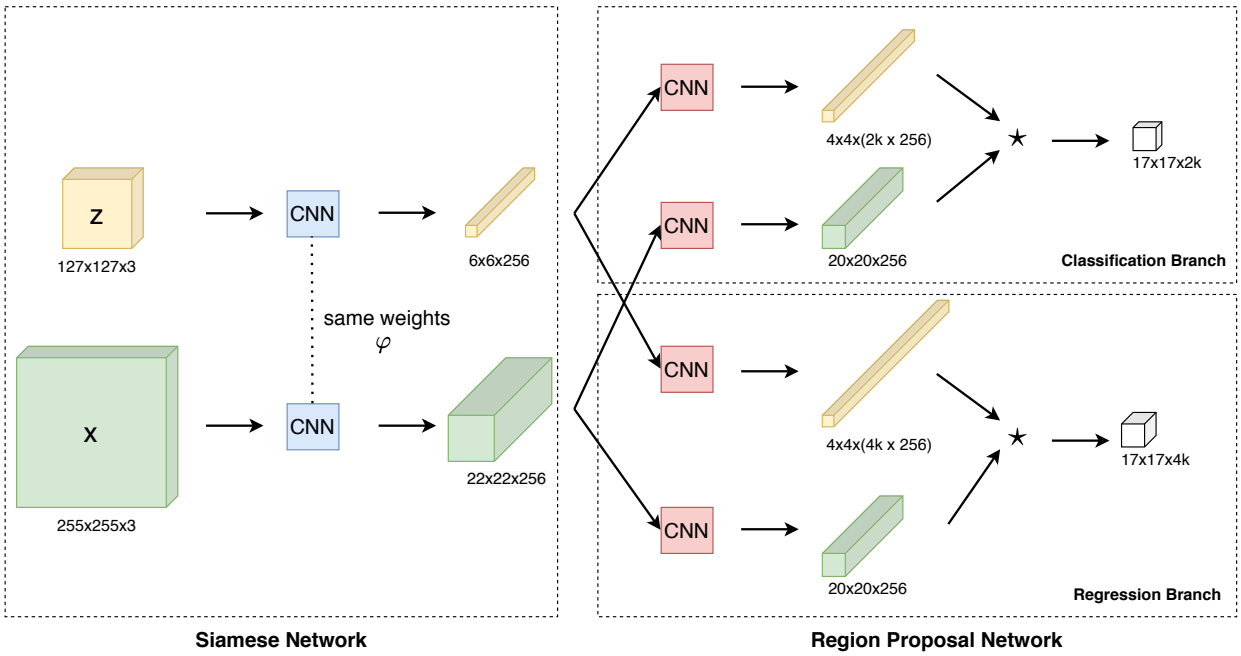


Figure 3: The architecture of Siamese-RPN.

from the object detection from video challenge in the 2015 edition of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [32]. The exemplar and search image of each pair are centered on the object and from the same video sequence with at most T other frames between them. The elements of the score map are considered positive if they are within radius R of the center, otherwise negative:

$$y[u] = \begin{cases} +1 & \text{if } k\|u - c\| \leq R \\ -1 & \text{otherwise,} \end{cases} \tag{8}$$

where u is the position of a score in the score map, c is the position of the center in the score map and k is the stride of the entire backbone neural networks. The logistic loss is used for each exemplar-candidate pair:

$$\ell(y, v) = \log(1 + \exp(-yv)), \tag{9}$$

where v is the real-valued score and $y \in \{+1, -1\}$ is the label based on Eqn. 8. Define the loss of a score map $\mathcal{D}$:

$$L(y, v) = \frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{D}} \ell(y[u], v[u]) \quad . \tag{10}$$

Denote the training set as T_{train}. Siamese Networks parameters θ are obtained by applying SGD to

$$\min_{\theta} \frac{1}{|T_{train}|} \sum_{(z,x,y) \in T_{train}} L(y, g(z, x; \theta)). \tag{11}$$

3.2. Advances in Siamese networks for visual object tracking

Limitations on Siamese-FC. Although Siamese-FC successfully applies Siamese networks to visual object tracking tasks achieving high accuracy and robustness at the time of its advent, it has some limitations shown as follows:

- Performing multi-scale search exhaustively to determine scale change of the object;
- Choosing 5 conv-layer AlexNet-like network as the backbone, which is shallow in the modern perspective;
- Only generating axis-aligned bounding box, which decreases IoU if the ground truth bounding boxes are not axis-aligned;
- Never updating the model during tracking, which hinders the chances of fine-tuning.

To overcome these limitations, many other Siamese networks based trackers have been proposed. Performing multi-scale tests in a brute-force manner significantly impedes the speed of the tracker. Siamese-RPN is proposed to add a Regional Proposal Network (RPN) [31] to automatically generate bounding boxes of different scales and aspect ratios [24]. Following Siamese-RPN, SiamRPN++ manages to employ backbones of more modern neural network architectures like ResNet [11, 23] to utilize richer feature representations. SiamMask tweaks the architecture of Siamese-RPN and adds a mask branch [38] to enable segmentation and predictions of non-axis bounding boxes.

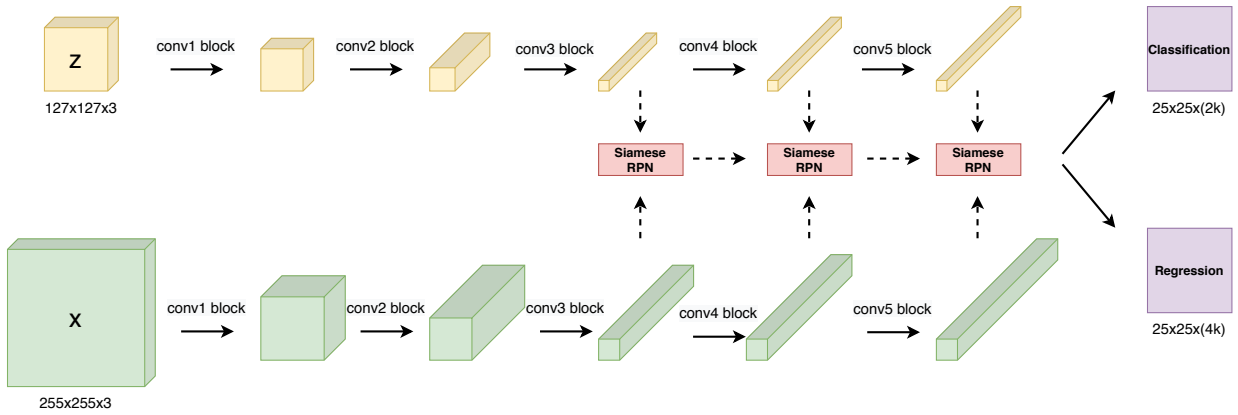


Figure 4: The architecture of SiamRPN++.

Siamese-RPN. Siamese-FC does not have bounding box regression but just brute-forcely searches for multiple scales. Besides, the aspect ratio of the object is always fixed. Siamese region proposal network (Siamese-RPN) solves this problem by introducing Regional Proposal Network (RPN), which is originated from the object detection task, to be integrated with Siamese networks [31, 24]. The architecture of Siamese-RPN is shown in Fig. 3. It consists of a Siamese subnetwork for feature extraction and a region proposal subnetwork for proposal refinement. Siamese subnetwork is similar to the counterpart of Siamese-FC. It takes as inputs an exemplar image of size $127 \times 127 \times 3$ and a search image of size $255 \times 255 \times 3$. The backbone is also an AlexNet-like network which has a slightly different architecture from the backbone of Siamese-FC. Hence, the feature embeddings of the inputs are of size $6 \times 6 \times 256$ and $22 \times 22 \times 256$, which are of the same spatial sizes of feature embeddings in Siamese-FC but are twice deeper. Region proposal subnetwork itself has two branches, one for classification and the other for regression. Denote the number of anchors for each spatial position in the feature embeddings as k. In the classification branch, the output feature map is of size $17 \times 17 \times 2k$, where $2k$ channels are due to the fact that foreground and background scores are calculated for each of the k anchors. In the regression branch, the output feature map is of size $17 \times 17 \times 4k$, where $4k$ channels corresponds to four coordinates dx, dy, dw, dh used for proposal refinement of each of the k anchors.

Inference in Siamese-RPN. Tracking in Siamese-RPN is also modeled as a local one-shot detection task. Since the exemplar image from the initial frame of each video sequence is fixed, the feature maps of size $4 \times 4 \times (2k \times 256)$ and $4 \times 4 \times (4k \times 256)$ are pre-computed and can be viewed as kernels to convolve with feature maps of the search image of each frame at inference time. The final feature maps of the classification branch and regression branch are used to determine the final proposal. Different from Siamese-FC, which directly chooses the bounding box corresponding to the maximum score, Siamese-RPN first chooses top K proposals and applies some strategies to determine the best proposal among them. The detailed procedures of deciding the predicted bounding box of the object of each frame at inference time are shown as follows:

1. Performing the forward pass with pre-computed kernels of the exemplar image to obtain the classification and regression feature maps.

2. Choosing top K points in the odd channel of the classification feature map.

3. Finding the corresponding 4 coordinate values of these top K points in the regression feature map and combining them with the anchor set to get the refined top K proposals.

4. Selecting the best proposal after applying two selection strategies:

   - discarding the bounding boxes generated by the anchors too far away from the center;
   - re-ranking the proposals by applying a cosine window and scale change penalty to scores.

Training Siamese-RPN. Like Siamese-FC, Siamese-RPN is trained on positive and negative pairs. Each pair is from the same video sequence in the training dataset derived from ILSVRC and Youtube-BB [32, 30]. While Siamese-FC uses distances between centers to determine the negativeness and positiveness of the pairs, Siamses-RPN adopted the IoU between anchors and the ground truth bounding box as the criterion which makes more sense as

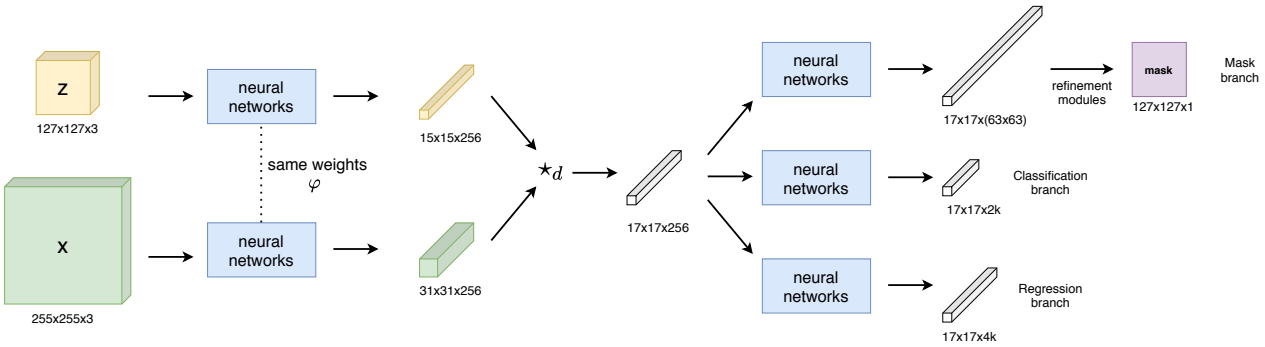


Figure 5: The architecture of SiamMask.

IoU is one of the main evaluating metrics. It allows the principles of two thresholds T_{high}, T_{low} to decide the label. Positive samples are defined as the pairs where the anchor and the ground truth bounding box have $IoU > T_{high}$; negative samples are defined as the pairs where the anchor and the ground truth bounding box have $IoU < T_{low}$. T_{high} is set to 0.6 and T_{low} is set to 0.3.

The loss function used in Siamese-RPN is the same as Faster R-CNN [31], which consists both parts of classification and regression:

$$L = \frac{1}{N_{cls}} \sum_i L_{cls,i} + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg,i}, \tag{12}$$

where p_i^* is 1 if the sample is positive and is 0 if the sample is negative. L_{cls} is the cross-entropy loss and L_{reg} is the smooth $L1$ loss of the normalized distance:

$$L_{reg} = \sum_{i=0}^{3} smooth_{L1}(\delta[i], \sigma). \tag{13}$$

Smooth $L1$ loss of independent variable x can be defined as:

$$smooth_{L1}(x, \sigma) = \begin{cases} \frac{1}{2}\sigma^2 x^2, & \text{if } |x| < \frac{1}{\sigma^2} \\ |x| - \frac{1}{2\sigma^2}, & \text{otherwise}, \end{cases} \tag{14}$$

which is proved to be less sensitive to outliers than the $L2$ loss in many vision tasks [10]. Let x_a, y_a, w_a, h_a denote center point and shape of the anchors, x, y, w, h denote those of the ground truth boxes, the normalized distance is:

$$\begin{aligned} \delta[0] = \frac{x - x_a}{w_a}, \delta[1] = \frac{y - y_a}{h_a}, \\ \delta[2] = \ln \frac{w}{w_a}, \ \delta[3] = \ln \frac{h}{h_a}. \end{aligned} \tag{15}$$

SiamRPN++. Both Siamese-FC and Siamese-RPN use modified AlexNet as the backbone, which has no padding and satisfies the full-convolutional property. With the developments of neural network architectures, many deeper and more complicated networks, such as ResNet [11], VGGNet [35], and MobileNetV2 [33], have achieved the state-of-the-art results in various other vision tasks like image recognition and object detection. However, modern neural networks like these must use padding to make the network deep, which violates the strict translation invariance property, i.e., fully-convolutional property. SiamRPN++ [23] is one of the first trackers that employ modern deep neural networks for visual object tracking, achieving state-of-the-art results. The authors hypothesize the violation of this restriction will lead to a spatial bias. They propose the so-called spatial aware sampling strategy which uses a shift of ± 64 pixel for training to alleviate the break of strict translation invariance property. The shift is defined as the max range of translation generated by a uniform distribution in data augmentation.

The architecture of SiamRPN++ is shown in Fig. 4. The ResNet-50 architecture is adopted with some refinements of the architecture to tailor it to the tracking task. ResNet-50 has 5 large residual blocks: $conv1$, $conv2$, $conv3$, $conv4$, and $conv5$. Each of these conv blocks, except $conv1$, contains multiple building blocks of skip connections and convolutional layers. Feature maps of earlier layers tend to capture low-level information while feature maps of latter layers can contain rich semantic information. By utilizing the fusion of feature maps of multiple blocks instead of just the final feature map, the tracker may have some performance gain. Specifically in SiamRPN++, the outputs of $conv3$, $conv4$ and $conv5$ are fed into three Siamese RPN modules, shown in red in Fig. 4, respectively. The outputs of the three RPN modules are of the same spatial size. Finally, the weighted sum of the three outputs is used for generating the final feature maps for classification and regression.

Due to the deep architecture, SiamRPN++ may have many more parameters which require more computation resources and decrease the tracking speed, compared to its predecessors. To solve this issue, SiamRPN++ is equipped with the Depthwise Cross-Correlation (DW-XCorr) Layer. This kind of layer performs cross correlation in a per chan-

nel way such that the output remains the same number of channels as the inputs. Applying this technique, the computational cost and memory usage are significantly reduced.

SiamMask. SiamMask proposes a unifying approach to perform fast online visual object tracking and segmentation at the same time. The architecture of SiamMask is shown in Fig. 5. Like SiamRPN++, it uses the modified ResNet-50 as the backbone of Siamese networks. For the cross correlation of the feature representations outputted from Siamese networks, it also employs the depth-wise cross correlation technique in SiamRPN++. The feature map of size $17 \times 17 \times 256$ is fed into three different branches for generating the response maps for classification, regression, and the binary mask. Since it has three branches, the loss function of SiamMask is a weighted sum of the classification loss, regression loss, and segmentation loss. The classification loss is the cross-entropy loss; the regression loss is the smooth $L1$ loss; the segmentation loss for the binary mask is the logistic regression loss over all responses of a candidate window (RoW).

Bounding box generation of SiamMask. In SiamMask, besides the axis-aligned bounding box generated from the classification and regression branch, the predicted bounding box as the tracking result can also be generated from the binary mask of the mask branch. By utilizing the binary mask, both axis-aligned and non-axis-aligned bounding boxes can be derived. Multiple strategies can be used. Axis-aligned bounding box (Min-max) is the axis-aligned bounding box of the minimum size that covers the maximum of the mask, i.e., the entire mask. Rotated minimum bounding rectangle (MBR) is the bounding box of the minimum size that covers the entire of the mask, which does not need to be axis-aligned. The third one is the optimization strategy (Opt) used for the automatic bounding box generation proposed in VOT2016 [17], which may not cover the entire mask. An example of these three strategies is shown in Fig. 6. The mask of the object is in yellow; the Min-max strategy is in red; the MBR strategy is in green; the Opt strategy is in blue.

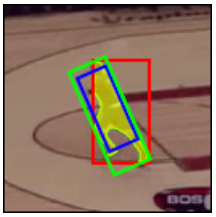


Figure 6: Three types of bounding box generation strategies.

4. Experiments

We evaluate the following series of Siamese Network based trackers: Siamese-FC, Siamese-RPN, SiamRPN++ (AlexNet, ResNet-50, and MobileNetV2 as backbones separately), and SiamMask. Interested readers can find the detailed results of those trackers comparing to other state-of-the-art trackers in the original papers [3, 24, 23, 38]. Here, we perform quantitative and qualitative analysis of the aforementioned trackers. Specifically, we give a visualization and evaluation of the score maps of Siamese-FC, analyze the performance of different backbone networks for several trackers and identifies bad cases when the trackers fail.

4.1. Experiments setup

System configurations. We run the experiments remotely on the Google Cloud Platform (GCP). The VM instance we use is configured based on Deep Learning VM Image provided in the marketplace of GCP. The node contains two Intel(R) Xeon(R) CPU @ 2.30GHz CPUs, running on Debian GNU/Linux (9) Operation System. The main memory is a 13GB RAM. The disk is a 200GB standard persistent disk of type `ext4`. Besides, one NVIDIA Tesla K80 GPU is employed.

Testing datasets. We analyze the trackers on several canonical benchmarks. One set of datasets we are using is from the VOT challenges. The VOT challenges [21] are a set of challenges for visual object tracking with the intention of creating canonical benchmarks and advancing trackers in multiple tracking tasks including short-term, long-term, and real-time tracking. The datasets we are using are the short-term tracking datasets, which include VOT2015, VOT2016, VOT2017/VOT2018, and VOT2019 dataset [20, 17, 16, 19]. One thing to notice is that VOT2018 is exactly the same dataset as VOT2017, so we represent this dataset as VOT2017/2018. Besides, these datasets are progressively more challenging: VOT2017/VOT2018 replaces the top 10 least challenging video sequences in VOT2016 and VOT2019 replaces 20% video sequences in VOT2017/2018. Another dataset we are using is the OTB2015 [41, 40] dataset, which contains 100 sequences from recent literature. Actually 98 different sequences are included in the dataset but two sequences, i.e., `Skating2` and `Jogging`, contain 2 objects and their corresponding ground-truth annotations.

Performance metrics. For evaluating results on VOT datasets, metrics in terms of both accuracy and robustness are used. Specifically, we are using accuracy, robustness, and expected average overlap (EAO) scores. EAO score, which is the main metric in the VOT challenges, measures

both accuracy and robustness [20, 21, 44]. A higher EAO score means better overall performance. Accuracy is the average intersection over union (IoU) while tracking. The IoU or sometimes called overlap ratio (OR) is defined as the ratio of the area of the intersection between the predicted and the ground-truth bounding boxes to the area of the union of the predicted and the ground-truth bounding boxes, whose value ranges from 0 to 1. Robustness is the failure rate that measures the number of times the tracker is reinitialized. When the output bounding box has zero overlap with the ground truth bounding box, the tracker fails at this frame and will be initialized again after 5 frames. We also use the number of losses to give a more direct measure for robustness.

For evaluating results on the OTB2015 dataset, multiple metrics in terms of accuracy are used. We do not reinitialize the tracker on OTB2015 if it fails. The two metrics we are using are the area under curve (AUC) of the success plot and the precision score of the precision plot. A success plot is the plot of success rate vs overlap ratio. A frame is successful if the OR of that frame is higher than some predefined threshold t and the success rate is the ratio of successful frames to all frames. By setting the threshold from 0 to 1, one can create the success plot of a tracker. Area under curve (AUC) is the area between the curve and x-axis of the success plot for x ranges from 0 to 1, which also ranges from 0 to 1 itself by definition. A precision plot shows the percentage of frames whose estimated center of the bounding box is within the given threshold distance of the ground truth center, i.e., the x-axis is the threshold (from 0 to 50 in convention) and the y-axis is the percentage of frames that satisfy the condition. The precision score is the percentage of frames that the Euclidean distance between the predicted center and the ground truth center is no more than 20.

4.2. Analysis of Siamese-FC

Visualization. We use the video sequence `Baskekball` which is a sequence of 725 frames in the OTB2015 dataset and also in the VOT datasets to visualize the mechanism of Siamese-FC. The sequence is an excerpt from a basketball game and the object to be tracked is a basketball player in green among several basketball players in either green or white. Each frame is of size 576×432. The first frame (frame 0) is shown in Fig. 7 with the ground truth bounding box of size 34×81. The ground truth bounding box of the frame is in red.

Siamese-FC will extract the exemplar image from the first frame for the entire sequence if no re-initialization is allowed. However, the exemplar image should be of a fixed size $127 \times 127 \times 3$. This can be achieved according to Eqn. 7. The exemplar image will not only include the ground truth bounding box, but also some context of the object. Fig. 8 shows the exemplar image of size $127 \times 127 \times 3$ for this sequence.

Figure 7: Frame 0 of `Basketball` with ground truth.

Figure 8: The exemplar image of `Basketball`.

(a) scale 1.037^{-1} (b) without scale (c) scale 1.037^{1}

Figure 9: Search images with different scales of frame 1 in `Basketball`.

For visualization of search images, we show the search images of frame 1 with three different scales in Fig. 9. All these images are of size $255 \times 255 \times 3$. Siamese-FC will find the corresponding position in the frame for the position achieving the highest score in the score map after applying a scaling penalty and a cosine window for large displacements. The corresponding score maps of size 272×272 are shown in Fig. 10. The lowest values are in the darkest black and the highest values are in the brightest white. The colors of the values go from black to red to yellow to white as their values go up. The score maps are in a very good radial shape and the highest value is near the center. In this case, the Siamese-FC has a good discriminative ability. Another good example is shown in Fig. 11 which is the score map of frame 10 in this sequence and the high values are in the shape of a human being.

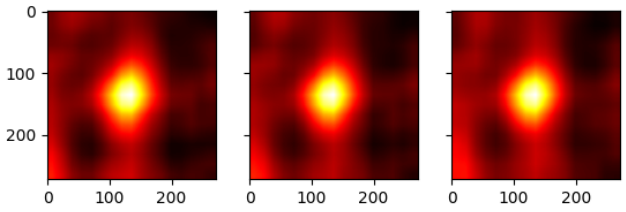


Figure 10: Score maps with different scales of frame 1 in `Basketball`.

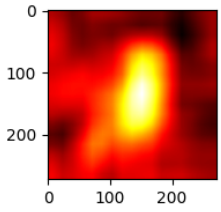
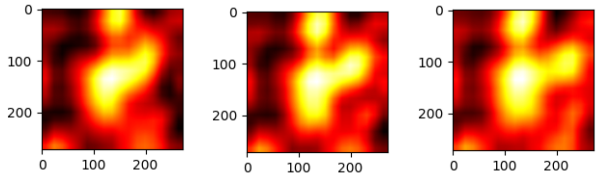


Figure 11: Score maps of frame 10 in `Basketball`.



(a) frame 473 (b) frame 474 (c) frame 475

Figure 12: Score maps of three consecutive frames in `Basketball`.

However, at about frame 474, a distractor, i.e., a different object similar to the target, comes and confuses Siamese-FC. The scores maps of three consecutive frames 473, 474, and 475 without scale are shown in Fig. 12. We can clearly see that there are multiple maximal in the score maps. Unfortunately, the distractor achieves the highest score which causes Siamese-FC tracks the distractor instead in subsequent frames. This result indicates that the discriminative ability of Siamese-FC can be further improved.

Selection strategy. In Siamese-FC, the center of the predicted bounding box is determined by the position of the max score in the score map. If multiple scales are tested at the same time, the scale that contains the highest score among all the score maps is chosen as the scale of the target object in this frame. The position of the highest score of that scale combined with the predicted center of the previous frame and current scale determines the predicted position of the center of the object in the current frame. However, choosing the position corresponding to the highest score may not yield the best performance.

We investigate another selection strategy: using the max score to determine which scale of the three different scales wins, then for that scale, we choose the second highest score as the better score and get the corresponding bounding box of it as the result of the model. The results on OTB2015 dataset are shown in Table. 1, where AO means average overlap ratio and precision is the percentage of the frames where the center of the predicted bounding box is within 20 pixels distance of the center of the ground truth bounding box. The upper arrow in the figure means a higher value of the metric shows better performance. Surprisingly, the AO has more than 0.01 gain and precision has more than 0.02 gain in our strategy. This proves that the similarity metric learned by Siamese-FC is not globally optimal, which implies that other selection strategies could be proposed to achieve better results.

Table 1: Results of a different selection strategy

strategy	AO↑	Precision↑
original	0.5947	0.7579
ours	0.6080	0.7780

Some other results on the visualization of score maps and the relation between scores and IoUs are included in Appendix A.

4.3. Analysis of backbone networks.

SiamRPN++ enables modern deep neural networks as the backbone architecture of Siamese networks. We evaluate SiamRPN++ with backbones of ResNet-50, MobileNetV2 as well as AlexNet. We also include the results of SiamMask, which has a backbone modified from ResNet-50.

The results on multiple VOT datasets are shown in Table 2. In the table, A stands for accuracy, R for robustness score, N for the number of loss and EAO for the expected average overlap. Higher accuracy, and EAO indicates better performance while a higher robustness score and number of loss means worse performance on robustness. This is also encoded in the table as up-arrow ↑, which means a higher value is better and down-arrow ↓, which means a lower value is better. The best results of each metric on each dataset are in bold font. Clearly, SiamRPN++ (AlexNet) achieves worst scores on all datasets in both accuracy and robustness metrics which conforms to the fact that SiamRPN++ (AlexNet) is the only tracker here which uses a relatively shallow architecture. SiamRPN++ (ResNet-50) achieves the best performance on VOT2016 and VOT2017/VOT2018 based on the EAO scores. The accuracy of all trackers on all the datasets is around 0.6, ranging from 0.572 to 0.642. However, the robustness score increases a lot from VOT2016 and VOT2017/VOT2018 to

Table 2: Results of 4 Siamese Network based trackers

	VOT2016	VOT2017/VOT2018	VOT2019
metrics	A↑/R↓/N↓/EAO↑	A↑/R↓/N↓/EAO↑	A↑/R↓/N↓/EAO↑
SiamRPN++ (AlexNet)	0.618/0.238/51/0.393	0.576/0.290/62/0.352	0.572/0.547/109/0.26
SiamRPN++ (ResNet-50)	**0.642/0.200/43/0.463**	**0.602**/0.234/50/**0.415**	0.594/0.467/93/0.287
SiamRPN++ (MobileNetV2)	0.624/0.214/46/0.455	0.586/**0.229/49**/0.410	0.580/**0.446/89/0.292**
SiamMask	0.620/0.214/46/0.436	0.597/0.248/53/0.406	**0.596**/0.467/93/0.283

Table 3: Average fps of 4 Siamese Network based trackers

fps	SiamRPN++ (AlexNet)	SiamRPN++ (ResNet-50)	SiamRPN++ (MobileNetV2)	SiamMask
official results	180	35	75	56
OTB2015	68.40	11.58	36.84	24.78
VOT2016	68.61	11.72	37.58	24.58
VOT2017/VOT2018	67.79	11.83	37.48	24.39
VOT2019	71.18	12.20	38.34	24.98

VOT2019. This means the trackers tend to fail, i.e., the predicted bounding box has zero overlap with the ground truth. One main reason for it is that VOT2019 replaces 12 easy video sequences in VOT2017/VOT2018 with video sequences of objects hard to track. The EAO score of each tracker also drops a lot on VOT2019 as it is a combination metric for both accuracy and robustness. The detailed number of losses for each tracker on each dataset is shown in Appendix B.

We also apply the trackers to the OTB2015 dataset. By the definition of area under curve (AUC) and the precision score, both scores indicate better performance of a tracker when they are of higher values. The results are shown in Table. 4. SiamRPN++ (ResNet-50) outperforms the other trackers in both AUC and precision. This matches with the results on VOT2016 and VOT2017/VOT2018. The detailed success score and precision score of each sequence for each tracker on OTB2015 can be found in Appendix B.

Table 4: Results on OTB2015

	AUC↑	Precision↑
SiamRPN++ (AlexNet)	0.648	0.853
SiamRPN++ (ResNet-50)	**0.667**	**0.880**
SiamRPN++ (MobileNetV2)	0.658	0.864
SiamMask	0.648	0.840

We not only want trackers to have good accuracy and robustness scores but also expect them to have a high speed, preferably in real-time. Otherwise, the scope of applications for the tracker is limited. The results of the fps of the trackers are shown in Table. 3. The official results are in the first row, which are 2x-2.5x times faster than our experiments. This matches with the fact that we are using Tesla K-80 GPU instead of GTX-1080Ti GPU, which is used in the original papers. From the table, we can see that SiamRPN++ (AlexNet) has the highest fps which is mainly due to the simplicity of its architecture. SiamRPN++ (ResNet-50) has the lowest fps but it is still a real-time tracker if running on a GPU of performance no worse than GTX-1080Ti. SiamRPN++ (MobileNetV2) is about twice faster than SiamRPN++ (ResNet-50) as its backbone is a light-weighted neural network based on an inverted residual structure for mobile applications. SiamMask has fps between SiamRPN++ with ResNet-50 and MobileNetV2. In real-world applications, the trackers can be chosen based on the tradeoff between the requirements of the performance and the limitations of the computing resources.

4.4. Analysis of failure cases

Trackers based on Siamese networks have good accuracy but the robustness may be greatly challenged in some special cases. For example, we can see from the experiment results of the previous subsection about the EAO scores of those trackers drops significantly on VOT2019. The main reason is that the trackers tend to fail multiple times on the newly-introduced video sequences in VOT2019. We find out the top 4 sequences that have the most number of losses: `agility`, `ball3`, `dribble`, and `hand2`, which happen to be among the new sequences in VOT2019. We identify that the losses are mainly due to these factors:

- large occlusion,
- motion blur,
- pose or viewpoint change,

- distractors.

The analysis with examples is included in Appendix C.

5. Conclusion

In this report, we provide a logical and elaborated description of the evolution of visual object tracking methods based on Siamese networks. We perform both quantitative and qualitative analysis of multiple representative Siamese networks based trackers like Siamese-FC, Siamese-RPN, SiamRPN++, and SiamMask. Concretely, we show how the score maps work and challenge the selection strategy in Siamese-FC. We analyze the effects of different backbone architectures in terms of accuracy, robustness, and speed. Based on the significant performance drop on the VOT2019 dataset, we also identify several factors that cause these trackers to fail.

Future directions. To facilitate future research, we propose here some potential directions to improve the performance of trackers based on Siamese networks. Direction one: exploring different selection strategies. As shown in the analysis, the simple selection strategies in the papers may not be optimal. We may even train a separate model to learn the optimal selection strategy. Direction two: exploring different architectures for Siamese networks. There are numerous neural networks in other computer vision tasks that achieve state-of-the-art results in their domains. Modifying those networks to tailor them for the tracking task and retraining them on the domain of tracking may work. Direction three: augmenting the training set. With fixed neural network architectures, the performance can still be improved if the training set is larger and properly augmented. Direction four: integrating on-line fine-tuning or latest correlation filters based methods to Siamese networks. Direction five: integrating Siamese networks with other models that achieve state-of-the-art results in other domains. For example, transformers [5] have recently achieved very good results in object detection and may be able to integrate Siamese networks in order to be applied to visual object tracking.

Acknowledgements. We would like to acknowledge Prof. Hu for his great support and assistance on this project, especially during this tough COVID-19 time. Prof. Hu's expertise in computer vision and signal processing gives invaluable insight into all aspects of the project. We also want to thank both senior graduate students and undergraduate students in the group for sharing research ideas and discussing various research topics.

References

[1] Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[2] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip H. S. Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 523–531, 2016.

[3] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In Gang Hua and Hervé Jégou, editors, *ECCV 2016 Workshops*, pages 850–865, Cham, 2016. Springer International Publishing.

[4] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2544–2550. IEEE Computer Society, 2010.

[5] Nicolas Carion, F. Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander M Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005.

[7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[12] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference Computer Vision (ECCV)*, 2016.

[13] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, abs/1404.7584, 2014.

[14] Yann LeCun Eduard Sickinger Jane Bromley, Isabelle Guyon and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems 6*, pages 737–744. 1993.

[15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Proceedings of the 32 nd International Conference on Machine Learning*, 2015.

[16] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomas Vojir, Gustav Häger, Alan Lukežič, Abdelrahman Eldesokey, and Gustavo Fernandez. The visual object tracking vot2017 challenge results. In *ICCV 2017 workshops*, 2017.

[17] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomas Vojir, Gustav Häger, Alan Lukežič, and Gustavo Fernandez. The visual object tracking vot2016 challenge results. In *ECCV 2016 Workshops*, Oct 2016.

[18] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pfugfelder, Luka Čehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, Gustavo Fernandez, and et al. The sixth visual object tracking vot2018 challenge results. In *ECCV 2018 Workshops*, 2018.

[19] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, Abdelrahman Eldesokey, Jani Kapyla, and Gustavo Fernandez. The seventh visual object tracking vot2019 challenge results. In *ICCV 2019 workshops*, 2019.

[20] Matej Kristan, Jiri Matas, Aleš Leonardis, Michael Felsberg, Luka Čehovin Zajc, Gustavo Fernandez, Tomas Vojir, Gustav Häger, Georg Nebehay, Roman Pflugfelder, Abhinav Gupta, Adel Bibi, Alan Lukežič, Alvaro Garcia-Martin, Amir Saffari, Alfredo Petrosino, and Andres Solis Montero. The visual object tracking vot2015 challenge results. In *ICCV 2015 Workshop*, pages 564–586, 2015.

[21] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, Nov 2016.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[23] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[26] Cristina Manresa, Javier Varona, Ramon Mas, and Francisco Perales. Hand tracking and gesture recognition for human-computer interaction. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 5(3):96–104, 2005.

[27] E. G. Miller, N. E. Matsakis, and P. A. Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 1, pages 464–471 vol.1, 2000.

[28] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4293–4302, June 2016.

[29] Junliang Xing Mengdan Zhang Weiming Hu Qiang Wang, Jin Gao. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017.

[30] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[34] A. Shukla and Mona Saini. "moving object tracking of vehicle detection": A concise review. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8:169–176, 03 2015.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[36] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014.

[37] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip H. S. Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[38] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.

[39] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors.

IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 34(3):334–352, 2004.

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

[41] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[42] Yinda Xu, Zeyu Wang, Zuoxin Li, Yuan Ye, and Gui Bo Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI 2020*, volume abs/1911.06188, 2019.

[43] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV 2018*, pages 103–119, 2018.

[44] L. Čehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. *IEEE Transactions on Image Processing*, 25(3):1261–1274, 2016.

Appendices

A Analysis of Siamese-FC

A.1 Visualization

We provide here more visualization of the score maps of Siamese-FC in some specific video sequences. In this part of the Appendix, the frame number starts at 1.

Synthetic Videos. In the OTB2015 data set, the `Bird1` sequence is a cartoon video which is created by computer graphics not from a camera. Even when the bird does not move too much, the tracker will still fail. The frame 1 is shown in Fig. 1.

Figure 1: Frame 1 of the `Bird1` sequence.

One example is the frame 40, where the wing has the highest score. The score map of this frame is shown in Fig. 2 and result in Fig. 3. This kind of case happens multiple times in the sequence.

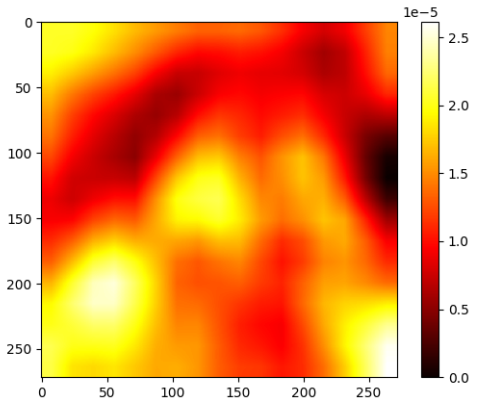


Figure 2: Score map of frame 40 of the `Bird1` sequence.

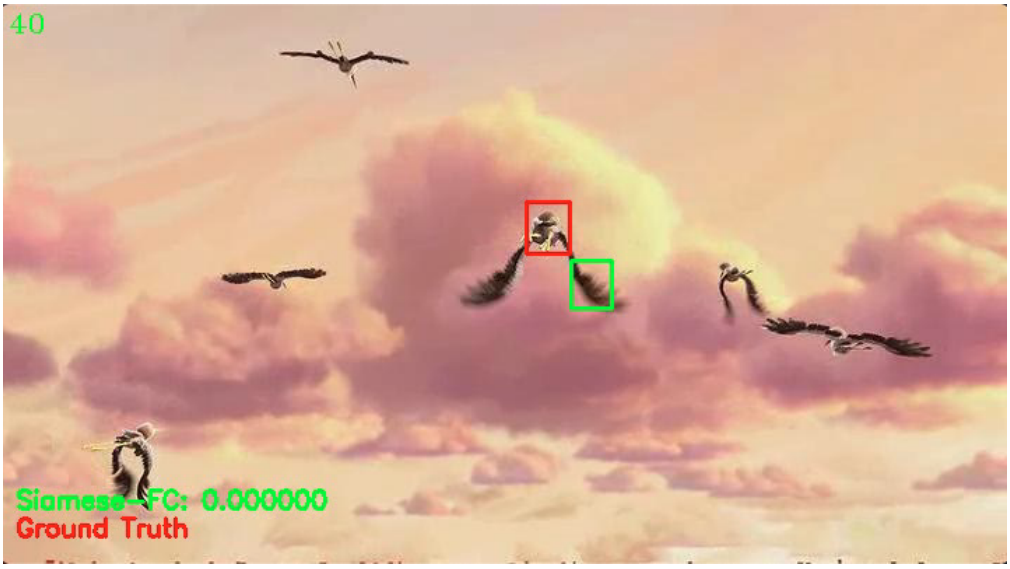
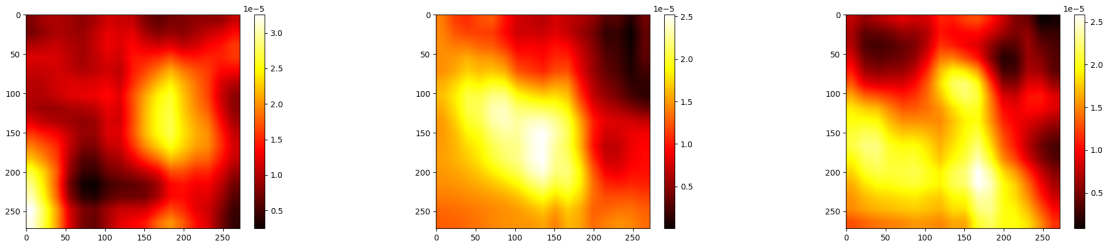


Figure 3: Frame 40 of the `Bird1` sequence.

Another issue of this video sequence is that during some period of time, the bird is completely occluded due to the cloud. The tracker also loses the object and the score maps look very weird. Some examples are shown in Fig. 4.



(a) Score map from frame 142 (b) Score map from frame 143 (c) Score map from frame 144

Figure 4: Score map of some frames when the object is completely occluded.

The result frame 143 is shown in Fig.5.

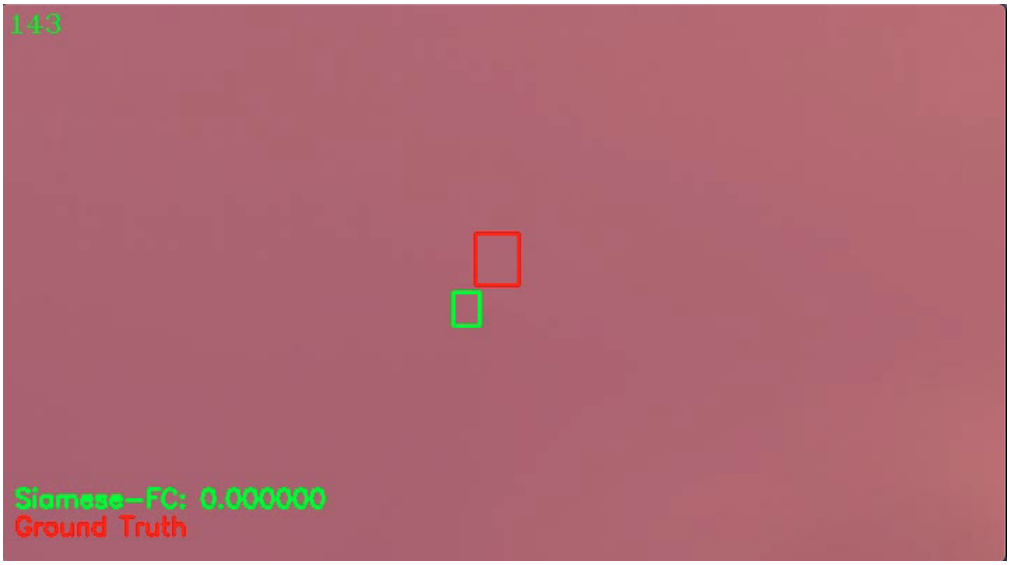


Figure 5: Frame 143 of the `Bird1` sequence.

Discrimination between object and context. Sometimes, the tracker keeps tracking of the context of the object in the first frame instead of the object. This illustrates that Siamese-FC is not discriminative enough to distinguish the object and its context in all cases. The sequence `Board` is an example, where a circuit board is moving over a table. However, the tracker is wandering around its initial position without tracking the real object.

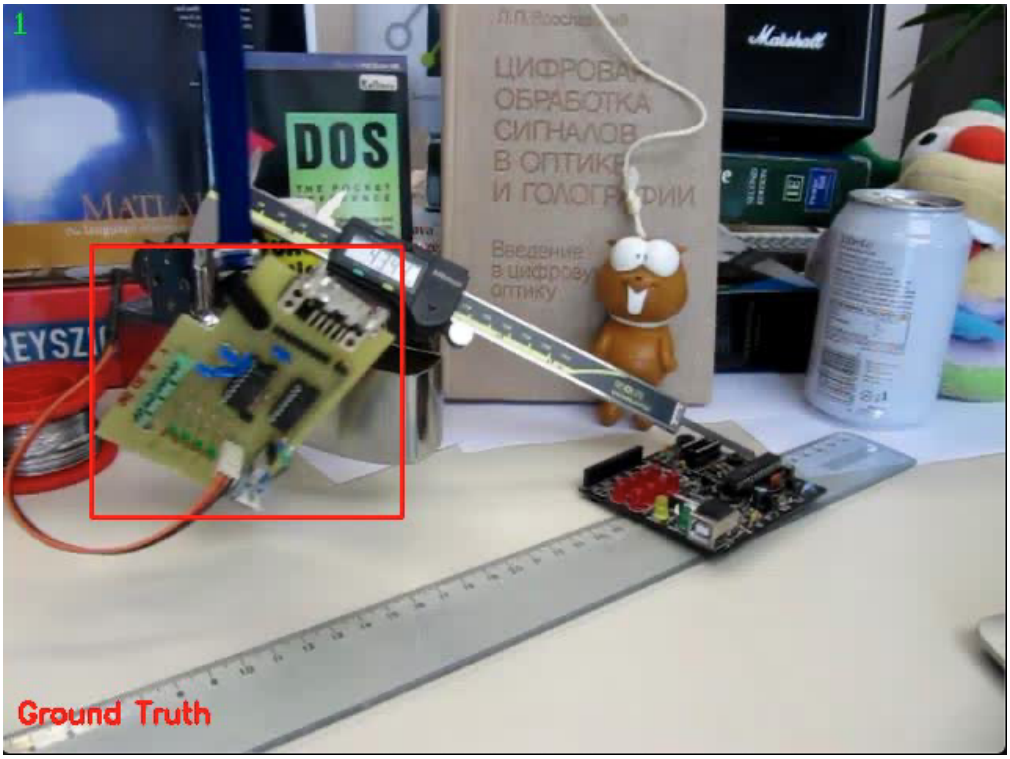


Figure 6: Frame 1 of the `Board` sequence.

In frame 28, the highest score in the score map is around the middle. However, according to the ground-truth, the highest score should at the right in the middle. The score map is shown in Fig. 7 and the result is shown in Fig. 8.

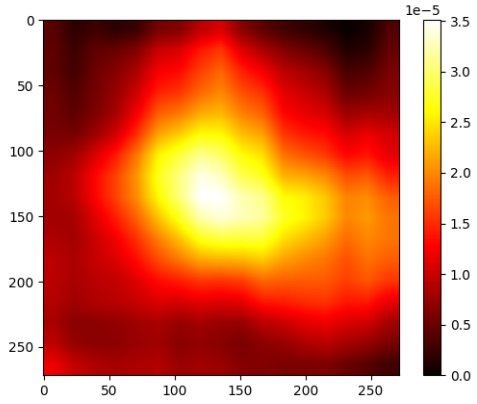


Figure 7: Score map of frame 28 of the `Board` sequence.

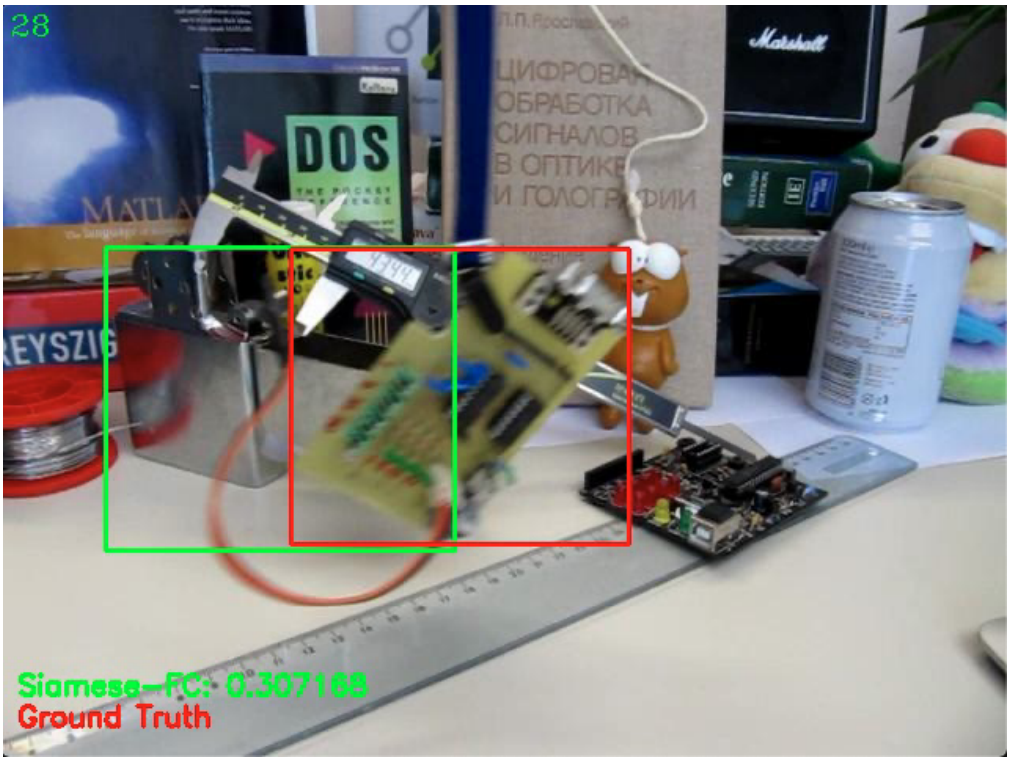
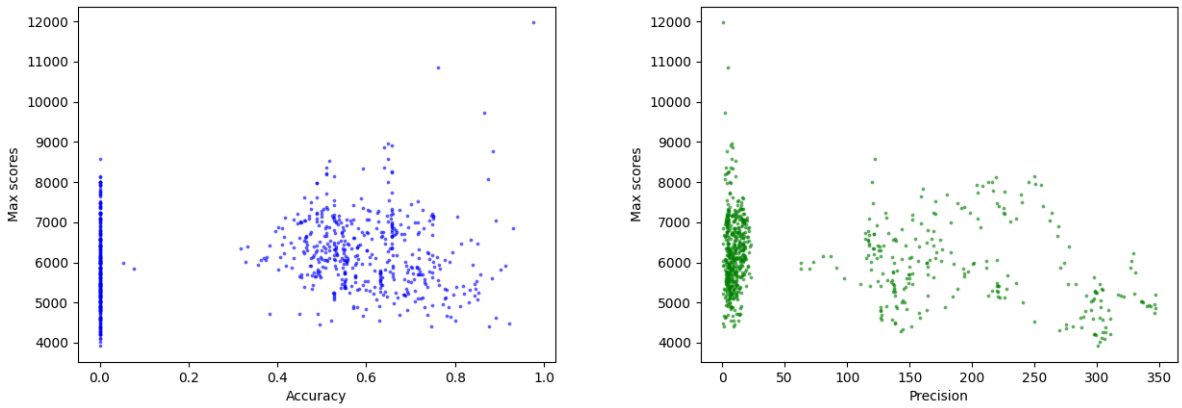


Figure 8: Frame 28 of the `Board` sequence.

A.2 Relationship between results

We use all 100 video sequences in OTB2015 to investigate the relation among accuracy, precision, and max scores in the score map. Accuracy is the IoU between the predicted bounding box and the ground truth bounding box; precision is the distance between the center of the predicted bounding box and the center of the ground truth bounding box.

Individual sequences. Fig. 9 shows the relation between accuracy (IoU) and the max score and the relation between precision and the max score in `Basketball` sequence. There is no clear relation between them.

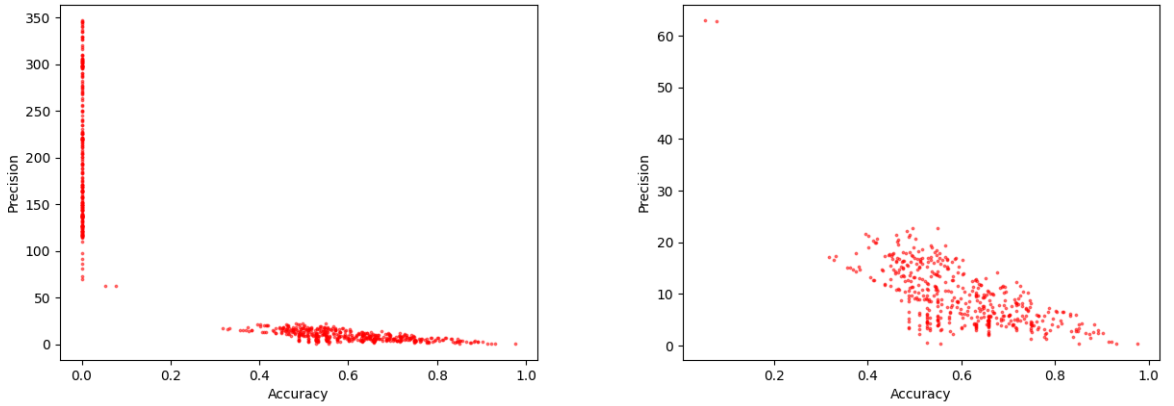


(a) max scores vs accuracy

(b) max scores vs precision

Figure 9: Relationship between metrics and max scores in `Basketball` sequence.

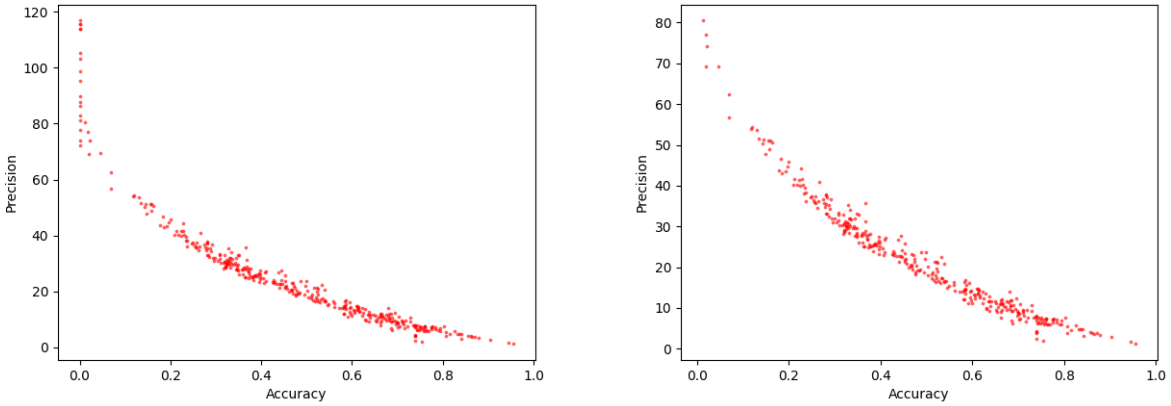
We also investigate the relation between the two metrics. We find out that they are actually somehow correlated. Fig. 10 shows the relation between the accuracy and precision of the `basketball` sequence. The relation is not clear. However, in some sequences, the relation is very clear. Our guess is that in those sequences, the scale of the object does not change too much so the relation between precision and accuracy is nearly quadratic. Fig. 11 shows one typical sequence `Tiger2`.



(a) precision vs accuracy

(b) precision ignoring zero accuracy vs accuracy

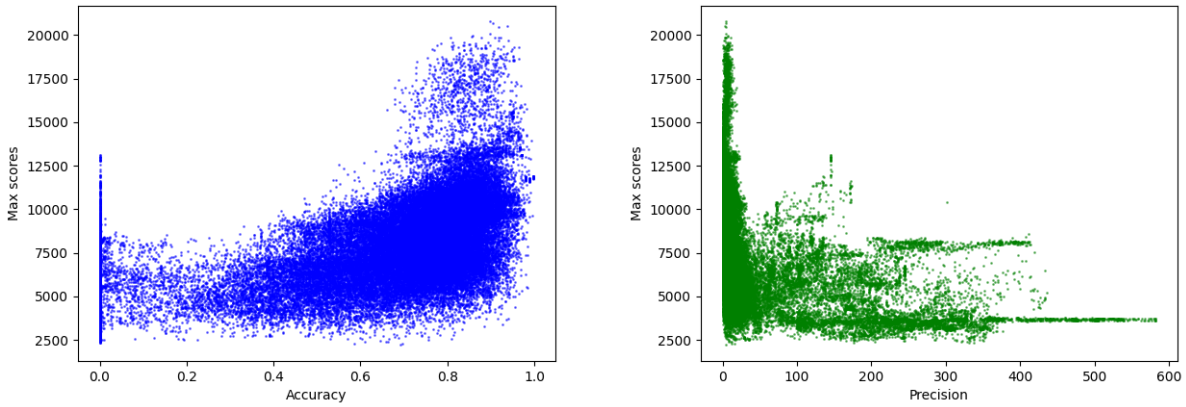
Figure 10: Relationship between metrics in `Basketball` sequence.



(a) precision vs accuracy (b) precision ignoring zero accuracy vs accuracy

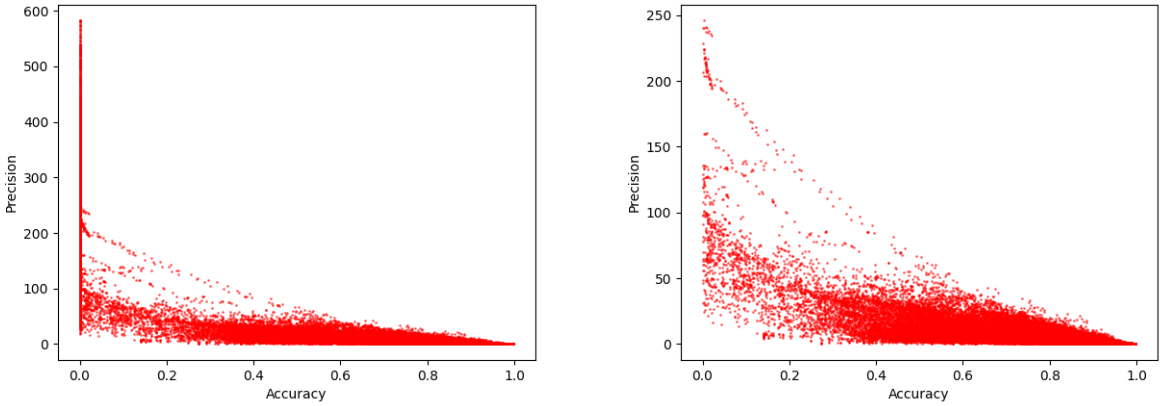
Figure 11: Relationship between metrics in `Tiger2` sequence.

The whole OTB2015 dataset. The same relationship analysis is applied to the entire OTB2015 dataset which contains 100 sequences. Fig. 12 shows the relationship between metrics and max scores in OTB2015 dataset and Fig. 13 shows the relationship between the 2 metrics. Like the findings in each individual sequence, there seems no clear relationship between accuracy and max score, as well as between precision and max score; but the relation between precision and accuracy also looks quadratic with different shapes as different sequences have different resolutions.



(a) max scores vs accuracy (b) max scores vs precision

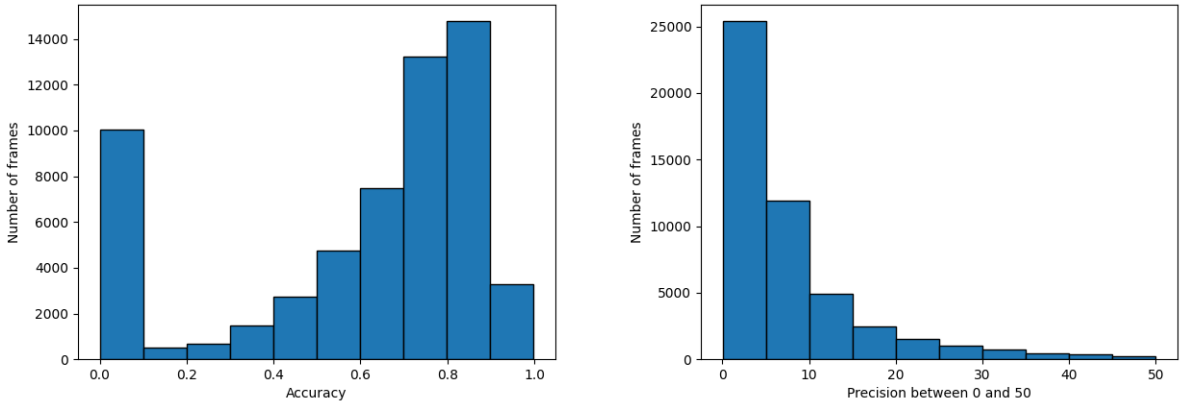
Figure 12: Relationship between metrics and max scores in OTB2015 dataset.



(a) precision vs accuracy (b) precision ignoring zero accuracy vs accuracy

Figure 13: Relationship of metrics in OTB2015 dataset.

We also include here the histogram of the accuracy and the precision of the whole dataset, which is shown in Fig. 14. The leftmost bar of the histogram of accuracy is mainly due to the fact that we do not enable re-detect: when the IoU is zero, no mechanism is employed to re-initialization the tracker. For successful tracking, most of the IoU lies between 0.7 and 0.9. For the histogram of precision, in most frames, the predicted center is no further than 20 Euclidean distance from the ground truth center.



(a) histogram of accuracy (b) histogram of precision between 0 and 50

Figure 14: Histogram of the metrics of the results in OTB2015.

B Analysis of backbone networks

We include the detailed number of losses for each video sequence on the three datasets VOT2016, VOT2017/VOT2018 and VOT2019 in Table. 1, Table. 2, and Table. 3 respectively.

Table 1: Detailed results on number of losses of 60 sequences on VOT2016

Sequences	SiamRPN++ (AlexNet)	SiamRPN++ (ResNet-50)	SiamRPN++ (MobileNetV2)	SiamMask
bag	0	0	0	0
ball1	0	0	0	0
ball2	1	0	0	0
basketball	2	1	1	2
birds1	2	2	2	0
birds2	0	0	0	0
blanket	0	1	1	2
bmx	1	0	1	0
bolt1	0	0	0	0
bolt2	0	1	1	0
book	2	1	2	5
butterfly	0	0	0	0
car1	1	1	1	1
car2	0	0	0	1
crossing	0	0	0	1
dinosaur	1	2	0	0
fernando	1	2	2	1
fish1	3	1	3	1
fish2	2	1	2	1
fish3	0	0	0	0
fish4	0	0	1	0
girl	3	4	2	3
glove	2	1	1	1
godfather	2	0	2	1
graduate	0	0	0	0
gymnastics1	0	0	0	2
gymnastics2	1	0	0	1
gymnastics3	1	1	1	1
gymnastics4	1	0	2	0
hand	0	1	0	0
handball1	0	2	1	0
handball2	2	2	1	3
helicopter	2	1	2	2
iceskater1	0	0	0	0
iceskater2	1	1	2	1
leaves	1	2	1	2
marching	0	0	0	0
matrix	1	0	0	0
motocross1	0	1	0	0
motocross2	0	0	0	0
nature	0	2	3	3
octopus	0	0	0	0
pedestrian1	0	0	0	0
pedestrian2	0	0	0	0
rabbit	4	3	5	2
racing	0	0	0	0
road	0	0	0	0
shaking	0	0	0	0
sheep	0	0	0	0
singer1	0	0	0	0
singer2	1	0	1	0
singer3	0	0	0	0
soccer1	8	5	3	4
soccer2	2	1	0	3
soldier	2	1	0	1
sphere	0	0	0	0
tiger	0	0	1	0
traffic	0	0	0	0
tunnel	0	0	0	0
wiper	1	2	1	1

Table 2: Detailed results on number of losses of 60 sequences on VOT2017/VOT2018

Sequences	SiamRPN++ (AlexNet)	SiamRPN++ (ResNet-50)	SiamRPN++ (MobileNetV2)	SiamMask
ants1	0	0	0	0
ants3	0	1	1	1
bag	0	0	0	0
ball1	0	0	0	0
ball2	1	0	0	0
basketball	2	1	1	2
birds1	1	0	0	1
blanket	0	1	1	2
bmx	0	0	1	0
bolt1	0	0	0	0
bolt2	0	1	1	0
book	2	1	2	5
butterfly	0	0	0	0
car1	1	0	1	0
conduction1	2	0	2	2
crabs1	2	3	1	1
crossing	0	0	0	1
dinosaur	1	2	0	0
drone1	2	3	0	1
drone_across	1	0	1	0
drone_flip	3	2	0	1
fernando	1	2	2	1
fish1	3	1	2	2
fish2	2	1	2	1
fish3	0	0	0	0
flamingo1	2	2	2	2
frisbee	2	1	3	1
girl	3	2	2	3
glove	2	1	1	1
godfather	2	1	2	1
graduate	0	0	0	0
gymnastics1	0	0	0	2
gymnastics2	1	0	0	1
gymnastics3	1	1	1	1
hand	0	1	0	0
handball1	0	2	1	0
handball2	2	3	3	1
helicopter	2	1	2	2
iceskater1	0	0	0	0
iceskater2	1	1	2	1
leaves	1	2	1	2
matrix	1	0	0	0
motocross1	1	1	0	0
motocross2	0	0	0	0
nature	2	3	2	2
pedestrian1	0	0	0	0
rabbit	4	3	5	2
racing	0	0	0	0
road	0	0	0	0
shaking	0	0	0	0
sheep	0	0	0	0
singer2	1	0	1	0
singer3	0	0	0	0
soccer1	5	3	1	4
soccer2	2	1	0	3
soldier	2	1	0	1
tiger	0	0	1	0
traffic	0	0	0	0
wiper	1	1	1	1
zebrafish1	0	0	0	1

Table 3: Detailed results on number of losses of 60 sequences on VOT2019

Sequences	SiamRPN++ (AlexNet)	SiamRPN++ (ResNet-50)	SiamRPN++ (MobileNetV2)	SiamMask
agility	5	5	4	5
ants1	0	0	0	0
ball2	1	0	0	0
ball3	18	17	16	15
basketball	2	1	1	2
birds1	1	0	0	1
bolt1	0	0	0	0
book	2	1	2	5
butterfly	0	0	0	0
car1	1	0	1	0
conduction1	2	0	2	2
crabs1	2	3	1	1
dinosaur	1	2	0	0
dribble	11	11	11	10
drone1	2	3	0	1
drone_across	1	0	1	0
drone_flip	3	2	0	1
fernando	1	2	2	1
fish1	3	1	2	2
fish2	2	1	2	1
flamingo1	2	2	2	2
frisbee	2	1	3	1
girl	3	2	2	3
glove	2	1	1	1
godfather	2	1	2	1
graduate	0	0	0	0
gymnastics1	0	0	0	2
gymnastics2	1	0	0	1
gymnastics3	1	0	1	1
hand	0	1	0	0
hand2	9	9	9	9
handball1	0	2	1	0
handball2	2	3	3	1
helicopter	2	1	2	2
iceskater1	0	0	0	0
iceskater2	1	1	2	1
lamb	1	1	1	1
leaves	1	2	1	2
marathon	3	2	2	2
matrix	1	0	0	0
monkey	0	1	1	0
motorcross1	1	1	0	0
nature	2	3	2	2
pedestrian1	0	0	0	0
polo	0	0	0	0
rabbit	4	3	5	2
rabbit2	0	0	0	0
road	0	0	0	0
rowing	0	0	0	2
shaking	0	0	0	0
singer2	1	0	1	0
singer3	0	0	0	0
soccer1	5	3	1	4
soccer2	2	1	0	3
soldier	2	1	0	1
surfing	0	0	0	0
tiger	0	0	1	0
wheel	0	0	0	0
wiper	1	1	1	1
zebrafish1	0	0	0	1

The detailed success score and precision score of each sequence for each tracker on OTB2015 is shown as follows.

Tracker name	siammask		siamrpn_alex		siamrpn_mobilev2		siamrpn_r50	
Video name	success	precision	success	precision	success	precision	success	precision
Basketball	0.512	0.653	0.620	0.858	0.022	0.030	0.404	0.559
Biker	0.721	0.943	0.402	0.463	0.742	0.944	0.738	0.940
Bird1	0.322	0.677	0.193	0.307	0.375	0.753	0.319	0.717
Bird2	0.696	0.876	0.650	0.749	0.686	0.807	0.682	0.823
BlurBody	0.789	0.866	0.816	0.853	0.785	0.862	0.816	0.870
BlurCar1	0.811	0.895	0.855	0.919	0.749	0.826	0.830	0.901
BlurCar2	0.859	0.906	0.823	0.901	0.838	0.917	0.835	0.920
BlurCar3	0.850	0.930	0.814	0.927	0.783	0.917	0.859	0.942
BlurCar4	0.867	0.889	0.866	0.908	0.843	0.864	0.871	0.903
BlurFace	0.777	0.832	0.784	0.875	0.631	0.836	0.754	0.882
BlurOwl	0.757	0.849	0.449	0.489	0.760	0.856	0.788	0.869
Board	0.609	0.480	0.326	0.270	0.759	0.673	0.272	0.205
Bolt	0.704	0.904	0.635	0.791	0.597	0.803	0.637	0.880
Bolt2	0.639	0.853	0.379	0.509	0.378	0.538	0.410	0.572
Box	0.781	0.870	0.747	0.846	0.800	0.894	0.801	0.880
Boy	0.791	0.945	0.801	0.943	0.815	0.948	0.806	0.949
Car1	0.816	0.967	0.742	0.959	0.768	0.962	0.818	0.967
Car2	0.842	0.943	0.809	0.938	0.836	0.944	0.832	0.943
Car24	0.819	0.952	0.841	0.951	0.848	0.956	0.859	0.956
Car4	0.856	0.952	0.857	0.947	0.842	0.940	0.855	0.949
CarDark	0.555	0.764	0.733	0.947	0.812	0.956	0.765	0.952
CarScale	0.830	0.922	0.843	0.936	0.826	0.928	0.846	0.938
ClifBar	0.413	0.627	0.508	0.728	0.231	0.299	0.506	0.725
Coke	0.648	0.870	0.562	0.817	0.597	0.855	0.680	0.844
Couple	0.746	0.915	0.730	0.920	0.757	0.920	0.721	0.919
Coupon	0.571	0.583	0.819	0.943	0.334	0.367	0.332	0.370
Crossing	0.798	0.958	0.793	0.957	0.812	0.963	0.800	0.959
Crowds	0.700	0.910	0.733	0.916	0.724	0.919	0.719	0.917
Dancer	0.696	0.833	0.757	0.864	0.742	0.840	0.727	0.826
Dancer2	0.763	0.841	0.753	0.809	0.788	0.854	0.733	0.826
David	0.623	0.877	0.685	0.898	0.513	0.893	0.480	0.901
David2	0.536	0.924	0.767	0.943	0.660	0.930	0.537	0.935
David3	0.751	0.905	0.704	0.866	0.721	0.895	0.691	0.912
Deer	0.589	0.811	0.605	0.818	0.524	0.801	0.509	0.747
Diving	0.417	0.477	0.562	0.762	0.302	0.378	0.373	0.482
Dog	0.469	0.853	0.435	0.856	0.480	0.839	0.521	0.868
Dog1	0.819	0.949	0.845	0.944	0.819	0.935	0.817	0.954
Doll	0.809	0.911	0.805	0.914	0.785	0.914	0.821	0.926
DragonBaby	0.501	0.575	0.695	0.841	0.658	0.758	0.663	0.795
Dudek	0.844	0.861	0.873	0.886	0.876	0.890	0.864	0.864
FaceOcc1	0.502	0.347	0.625	0.557	0.588	0.477	0.573	0.417
FaceOcc2	0.413	0.687	0.655	0.834	0.575	0.825	0.528	0.814
Fish	0.848	0.926	0.805	0.929	0.833	0.915	0.806	0.925
FleetFace	0.602	0.602	0.733	0.736	0.749	0.740	0.781	0.798
Football	0.737	0.913	0.243	0.317	0.261	0.330	0.281	0.386
Football1	0.762	0.931	0.625	0.880	0.779	0.937	0.808	0.941
Freeman1	0.665	0.896	0.705	0.897	0.672	0.883	0.656	0.875
Freeman3	0.790	0.959	0.822	0.958	0.826	0.962	0.817	0.960
Freeman4	0.391	0.539	0.362	0.478	0.692	0.941	0.680	0.936
Girl	0.722	0.906	0.719	0.925	0.776	0.931	0.732	0.893
Girl2	0.369	0.384	0.407	0.472	0.460	0.546	0.556	0.652
Gym	0.663	0.856	0.547	0.831	0.668	0.848	0.636	0.859
Human2	0.751	0.778	0.767	0.788	0.786	0.801	0.779	0.805
Human3	0.015	0.019	0.036	0.047	0.048	0.060	0.038	0.048
Human4-2	0.355	0.489	0.392	0.475	0.603	0.765	0.699	0.914
Human5	0.781	0.925	0.753	0.917	0.738	0.909	0.770	0.931
Human6	0.774	0.873	0.680	0.803	0.757	0.879	0.750	0.881
Human7	0.819	0.933	0.830	0.942	0.831	0.945	0.818	0.943
Human8	0.846	0.959	0.779	0.932	0.824	0.961	0.795	0.959
Human9	0.822	0.951	0.772	0.946	0.792	0.941	0.804	0.956
Ironman	0.234	0.286	0.516	0.638	0.585	0.719	0.464	0.578
Jogging-1	0.158	0.224	0.650	0.876	0.502	0.737	0.629	0.890
Jogging-2	0.695	0.871	0.676	0.868	0.709	0.883	0.730	0.937

Jump	0.140	0.105	0.241	0.350	0.346	0.473	0.240	0.276
Jumping	0.694	0.922	0.655	0.926	0.716	0.921	0.650	0.895
KiteSurf	0.773	0.951	0.773	0.939	0.628	0.840	0.726	0.871
Lemming	0.764	0.864	0.773	0.886	0.774	0.866	0.657	0.736
Liquor	0.436	0.478	0.453	0.473	0.600	0.669	0.476	0.523
Man	0.792	0.942	0.837	0.959	0.856	0.961	0.862	0.962
Matrix	0.599	0.825	0.483	0.631	0.299	0.392	0.727	0.868
Mhyang	0.816	0.927	0.746	0.906	0.725	0.940	0.754	0.931
MotorRolling	0.721	0.813	0.696	0.782	0.787	0.866	0.651	0.721
MountainBike	0.768	0.897	0.743	0.889	0.792	0.912	0.845	0.926
Panda	0.630	0.926	0.628	0.928	0.631	0.927	0.625	0.926
RedTeam	0.684	0.948	0.723	0.947	0.704	0.942	0.739	0.946
Rubik	0.723	0.773	0.673	0.735	0.746	0.814	0.723	0.761
Shaking	0.793	0.891	0.761	0.870	0.799	0.904	0.821	0.912
Singer1	0.750	0.878	0.737	0.909	0.728	0.842	0.767	0.844
Singer2	0.604	0.514	0.553	0.633	0.686	0.783	0.636	0.594
Skater	0.760	0.870	0.740	0.858	0.778	0.884	0.732	0.862
Skater2	0.695	0.850	0.682	0.841	0.702	0.825	0.733	0.865
Skating1	0.446	0.553	0.510	0.639	0.732	0.875	0.491	0.615
Skating2-1	0.430	0.526	0.404	0.511	0.449	0.553	0.464	0.558
Skating2-2	0.451	0.442	0.539	0.530	0.466	0.449	0.464	0.452
Skiing	0.597	0.922	0.569	0.918	0.622	0.936	0.630	0.930
Soccer	0.175	0.207	0.329	0.497	0.324	0.445	0.406	0.623
Subway	0.790	0.939	0.727	0.931	0.770	0.935	0.808	0.949
Surfer	0.702	0.912	0.640	0.864	0.519	0.709	0.674	0.905
Suv	0.404	0.535	0.592	0.744	0.581	0.711	0.654	0.768
Sylvester	0.786	0.913	0.723	0.898	0.736	0.905	0.767	0.903
Tiger1	0.641	0.713	0.628	0.701	0.644	0.701	0.663	0.744
Tiger2	0.637	0.724	0.632	0.720	0.643	0.720	0.631	0.721
Toy	0.553	0.754	0.713	0.825	0.668	0.838	0.644	0.794
Trans	0.693	0.626	0.620	0.369	0.658	0.457	0.741	0.730
Trellis	0.758	0.877	0.691	0.878	0.672	0.911	0.699	0.916
Twinnings	0.711	0.936	0.674	0.871	0.728	0.924	0.714	0.917
Vase	0.575	0.680	0.569	0.685	0.611	0.733	0.581	0.716
Walking	0.769	0.945	0.690	0.949	0.755	0.945	0.751	0.947
Walking2	0.296	0.377	0.282	0.454	0.360	0.542	0.466	0.599
Woman	0.648	0.897	0.623	0.867	0.658	0.913	0.616	0.902

C Analysis of failure cases

We illustrate here the top 4 sequences that have the most number of losses: `agility`, `ball3`, `dribble`, and `hand2`, which happen to be among the new sequences in VOT2019. In the following demonstration, the bounding boxes of the ground truth are all in red, those of SiamMask are in green, those of SiamRPN++ (Alex) are in blue, and those of SiamRPN++ (ResNet-50) are in yellow. The bounding boxes of the ground truth are often not axis-aligned. The bounding boxes of SiamRPN++ are always axis-aligned due to the nature of the tracker while the bounding box of SiamMask can be either axis-aligned or not axis-aligned. In this part of the Appendix, the frame number starts at 1.

C.1 Large occlusion

We demonstrate large occlusion using `agility`, which contains 100 frames. All the 3 trackers have 5 losses on the sequence. Fig. 15 shows the first frame of the sequence where every tracker gets initialized. The trackers work well with small occlusions, like in frame 10 shown in Fig. 16. However, with large or total occlusion, the trackers fail, like in frame 17 shown in Fig. 17 and frame 76 in Fig. 18. One thing to notice is that the ground truth bounding box counts for not only the visible part of the object but also its occluded part, which makes it even harder to estimate a precise bounding box for the object.

Figure 15: Frame 1 of agility.

Figure 16: Frame 10 of agility.

Figure 17: Frame 17 of agility.

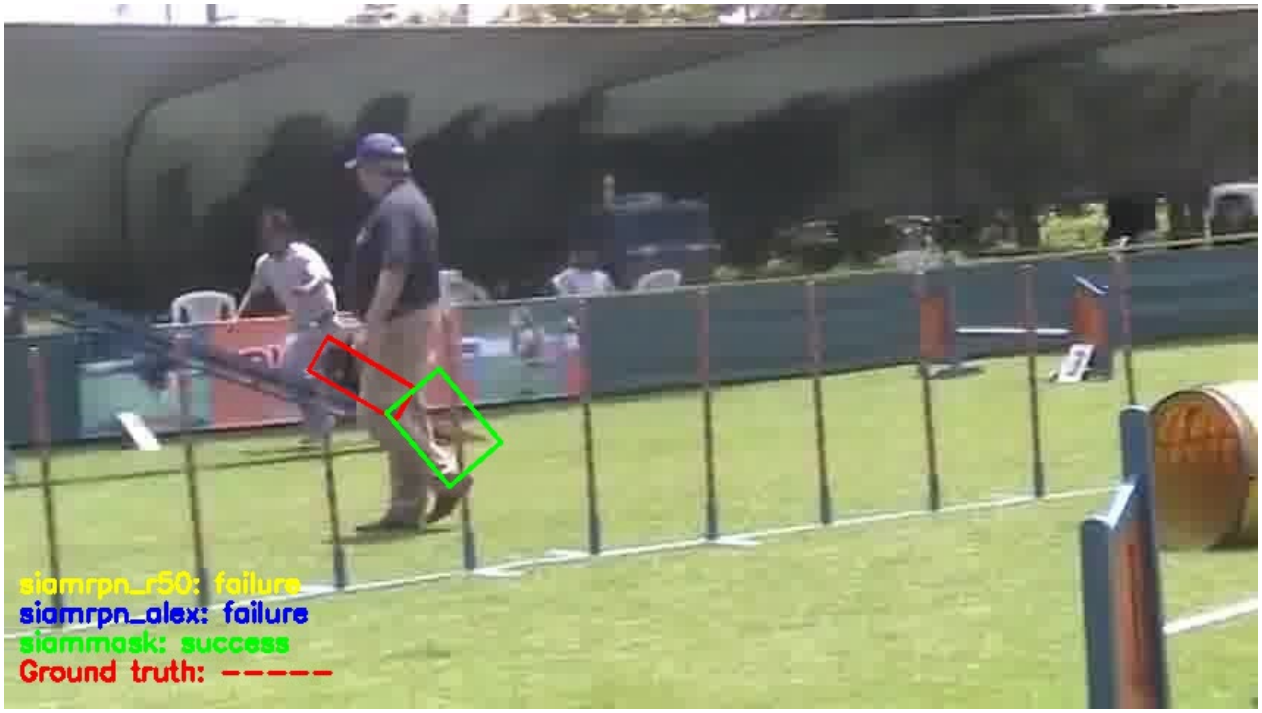


Figure 18: Frame 76 of agility.

C.2 Motion blur

We demonstrate motion blur using `ball3`, which contains 171 frames. All the 3 trackers have at least 15 losses on the sequence. The object, which is a ball, moves very fast and frequently alternates directions in the video sequence. The first frame and one success tracking are shown in Fig. 19. The failure due to the

motion blur of fast movement is shown in Fig. 20.

(a) Frame 1 of ball3 (b) Frame 5 of ball3

Figure 19: Initialization and a successful frame in ball3.

(a) Frame 9 of ball3 (b) Frame 115 of ball3

Figure 20: Some failures in ball3.

C.3 Pose and viewpoint change

We demonstrate the effect of pose and viewpoint using `hand2`, which contains 145 frames. All the 3 trackers have 9 losses on the sequence. The object of the sequence is a hand that can have various poses. The first frame and one success tracking frame are shown in Fig. 21 and in Fig. 22 respectively.

Figure 21: Frame 1 of hand2.

Figure 22: Frame 12 of hand2.

However, the trackers start to fail when the post changes, sometimes with the combination of fast movement. The small size of the hand makes the problem even severer. One example is shown in Fig. 23. And they fail in the next frame, which is shown in Fig. 24.

Figure 23: Frame 28 of hand2.

Figure 24: Frame 29 of hand2.

C.4 Distractors

One case of distractors is shown in the main text of the report when the Siamese-FC is analyzed. Details are omitted here.