SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks

Bo Li*
SenseTime Research
libo@sensetime.com

Wei Wu*
SenseTime Research
wuwei@sensetime.com

Qiang Wang*
NLPR, CASIA
qiang.wang@nlpr.ia.ac.cn

Fangyi Zhang
VIPL, ICT
fangyi.zhang@vipl.ict.ac.cn

Junliang Xing
NLPR, CASIA
jlxing@nlpr.ia.ac.cn

Junjie Yan
SenseTime Research
yanjunjie@sensetime.com

Abstract

Siamese network based trackers formulate tracking as convolutional feature cross-correlation between a target template and a search region. However, Siamese trackers still have an accuracy gap compared with state-of-the-art algorithms and they cannot take advantage of features from deep networks, such as ResNet-50 or deeper. In this work we prove the core reason comes from the lack of strict translation invariance. By comprehensive theoretical analysis and experimental validations, we break this restriction through a simple yet effective spatial aware sampling strategy and successfully train a ResNet-driven Siamese tracker with significant performance gain. Moreover, we propose a new model architecture to perform layer-wise and depth-wise aggregations, which not only further improves the accuracy but also reduces the model size. We conduct extensive ablation studies to demonstrate the effectiveness of the proposed tracker, which obtains currently the best results on five large tracking benchmarks, including OTB2015, VOT2018, UAV123, LaSOT, and TrackingNet.

1. Introduction

Visual object tracking has received increasing attention over the last decades and has remained a very active research direction. It has a large range of applications in diverse fields like visual surveillance [49], human-computer interactions [27], and augmented reality [50]. Although much progress has been made recently, it has still been commonly recognized as a very challenging task due to numerous factors such as illumination variation, occlusion, and background clutters, to name a few [48].

Recently, the Siamese network based trackers [42, 1, 16, 44, 43, 25, 45, 54, 46] have drawn much attention in the community. These Siamese trackers formulate the visual object tracking problem as learning a general similarity map by cross-correlation between the feature representations learned for the target template and the search region. To ensure tracking efficiency, the offline learned Siamese similarity function is often fixed during the running time [42, 1, 16]. The CFNet tracker [43] and DSiam tracker [12] update the tracking model via a running average template and a fast transformation module, respectively. The SiamRNN tracker [25] introduces the region proposal network [25] after the Siamese network and performs joint classification and regression for tracking. The DaSiamRPN tracker [54] further introduces a distractor-aware module and improves the discrimination power of the model.

Although the above Siamese trackers have obtained outstanding tracking performance, especially for the well-balanced accuracy and speed, even the best performed Siamese trackers, such as SiamPRN, the accuracy still has a notable gap with the state-of-the-arts [5] on tracking benchmarks like OTB2015 [48]. We observe that all these trackers have built their network upon architecture similar to AlexNet [24] and tried several times to train a Siamese tracker with more sophisticated architecture like ResNet [15] yet with no performance gain. Inspired by this observation, we perform an analysis of existing Siamese trackers and find the core reason comes from the destroy of the strict translation invariance. Since the target may appear at any position in the search region, the learned feature representation for the target template should stay spatial invariant, and we further theoretically find that, among modern deep architectures, only the zero-padding variant of AlexNet satisfies this spatial invariance restriction.

To overcome this restriction and drive the Siamese tracker with more powerful deep architectures, through extensive experimental validations, we introduce a simple yet effective sampling strategy to break the spatial invariance restriction of the Siamese tracker. We successfully

*The first three authors contributed equally. Work done at SenseTime. Project page: http://bo-li.info/SiamRPN++.

train a SiamRPN [25] based tracker using the ResNet as a backbone network and obtain significant performance improvements. Benefiting from the ResNet architecture, we propose a layer-wise feature aggravation structure for the cross-correlation operation, which helps the tracker to predict the similarity map from features learned at multiple levels. By analyzing the Siamese network structure for cross-correlations, we find that its two network branches are highly imbalanced in terms of parameter number; therefore we further propose a depth-wise separable correlation structure which not only greatly reduces the parameter number in the target template branch, but also stabilizes the training procedure of the whole model. In addition, an interesting phenomena is observed that objects in the same categories have high response on the same channels while responses of the rest channels are suppressed. The orthogonal property may also improve the tracking performance.

To summarize, the main contributions of this work are listed below in fourfold:

- We provide a deep analysis of Siamese trackers and prove that when using deep networks the decrease in accuracy comes from the destroy of the strict translation invariance.
- We present a simple yet effective sampling strategy to break the spatial invariance restriction which successfully trains Siamese tracker driven by a ResNet architecture.
- We propose a layer wise feature aggregation structure for the cross-correlation operation, which helps the tracker to predict the similarity map from features learned at multiple levels.
- We propose a depth-wise separable correlation structure to enhance the cross-correlation to produce multiple similarity maps associated with different semantic meanings.

Based on the above theoretical analysis and technical contributions, we have developed a highly effective and efficient visual tracking model which establishes a new state-of-the-art in terms of tracking accuracy, while running efficiently at 35 FPS. The proposed tracker, referred as *SiamRPN++*, consistently obtains the best tracking results on five of the largest tracking benchmarks, including OTB2015 [48], VOT2018 [22], UAV123 [32], LaSOT [10], and TrackingNet [31]. Furthermore, we propose a fast variant of our tracker using MobileNet[19] backbone that maintains competitive performance, while running at 70 FPS. To facilitate further studies on the visual tracking direction, we will release the source code and trained models of the SiamRPN++ tracker.

2. Related Work

In this section, we briefly introduce recent trackers, with a special focus on the Siamese trackers [42, 1]. Besides, we also describe the recent developments of deep architectures.

Visual tracking has witnessed a rapid boost in the last decade due to the construction of new benchmark datasets [47, 48, 20, 22, 10, 31] and improved methodologies [17, 53, 6, 7, 18, 33, 9, 5, 45, 54, 51]. The standardized benchmarks [47, 48, 10] provide fair testbeds for comparisons with different algorithms. The annually held tracking challenges [23, 20, 21, 22] are consistently pushing forward the tracking performance. With these advancements, many promising tracking algorithms have been proposed. The seminal work by Bolme *et al*. [3] introduces the Convolution Theorem from the signal processing field into visual tracking and transforms the object template matching problem into a correlation operation in the frequency domain. Own to this transformation, the correlation filter based trackers gain not only highly efficient running speed, but also increase accuracy if proper features are used [17, 52, 53, 8, 6]. With the wide adoption of deep learning models in visual tracking, tracking algorithms based on correlation filter with deep feature representations [9, 5] have obtained the state-of-the-art accuracy in popular tracking benchmarks [47, 48] and challenge [23, 20, 21].

Recently, the Siamese network based trackers have received significant attentions for their well-balanced tracking accuracy and efficiency [42, 1, 16, 44, 43, 13, 25, 45, 54, 46]. These trackers formulate visual tracking as a cross-correlation problem and are expected to better leverage the merits of deep networks from end-to-end learning. In order to produce a similarity map from cross-correlation of the two branches, they train a Y-shaped neural network that joins two network branches, one for the object template and the other for the search region. Additionally, these two branches can remain fixed during the tracking phase [42, 1, 16, 45, 25, 54] or updated online to adapt the appearance changes of the target [44, 43, 13]. The currently state-of-the-art Siamese trackers [25, 54] enhance the tracking performance by a region proposal network after the Siamese network and produce very promising results. However, on the OTB benchmark [48], their tracking accuracy still leaves a relatively large gap with state-of-the-art deep trackers like ECO [5] and MDNet [33].

With the proposal of modern deep architecture AlexNet by Krizhevsky *et al*. [24] in 2012, the studies of the network architectures are rapidly growing and many sophisticated deep architectures are proposed, such as VGGNet [38], GoogleNet [39], ResNet [15] and MobileNet [19]. These deep architectures not only provide deeper understanding on the design of neural networks, but also push forwards the state-of-the-arts of many computer vision tasks like object detection [34], image segmentation [4], and human pose estimation [40]. In deep visual trackers, the network architecture usually contains no more than five constitutional layers tailored from AlexNet or VGGNet. This phenomenon is explained that shallow features mostly contribute to the accurate localization of the object [35]. In this work, we argue that the performance of Siamese trackers can significantly get boosted using deeper models if the model is properly trained with the whole Siamese network.

3. Siamese Tracking with Very Deep Networks

The most important finding of this work is that the performance of the Siamese network based tracking algorithm can be significantly boosted if it is armed with much deeper networks. However, simply training a Siamese tracker by directly using deeper networks like ResNet does not obtain the expected performance improvement. We find the underlying reason largely involves the intrinsic restrictions of the Siamese trackers, Therefore, before the introduction of the proposed SiamRPN++ model, we first give a deeper analysis on the Siamese networks for tracking.

3.1. Analysis on Siamese Networks for Tracking

The Siamese network based tracking algorithms [42, 1] formulate visual tracking as a cross-correlation problem and learn a tracking similarity map from deep models with a Siamese network structure, one branch for learning the feature presentation of the target, and the other one for the search area. The target patch is usually given in the first frame of the sequence and can be viewed as an exemplar $\mathbf{z}$. The goal is to find the most similar patch (instance) from following frame $\mathbf{x}$ in a semantic embedding space $\phi(\cdot)$:

$$f(\mathbf{z}, \mathbf{x}) = \phi(\mathbf{z}) * \phi(\mathbf{x}) + b, \qquad (1)$$

where b is used to model the offset of the similarity value.

This simple matching function naturally implies two *intrinsic* restrictions in designing a Siamese tracker.

- The contracting part and the feature extractor used in Siamese trackers have an intrinsic restriction for *strict translation invariance*, $f(\mathbf{z}, \mathbf{x}[\triangle\tau_j]) = f(\mathbf{z}, \mathbf{x})[\triangle\tau_j]$, where $[\triangle\tau_j]$ is the translation shift sub window operator, which ensures the efficient training and inference.
- The contracting part has an intrinsic restriction for *structure symmetry*, *i.e.* $f(\mathbf{z}, \mathbf{x}') = f(\mathbf{x}', \mathbf{z})$, which is appropriate for the similarity learning.

After detailed analysis, we find the core reason for preventing Siamese tracker using deep network is related to these two aspects. Concretely speaking, one reason is that padding in deep networks will destroy the strict translation invariance. The other one is that RPN requires *asymmetrical* features for classification and regression. We will introduce spatial aware sampling strategy to overcome the first problem, and discuss the second problem in Sect. 3.4.

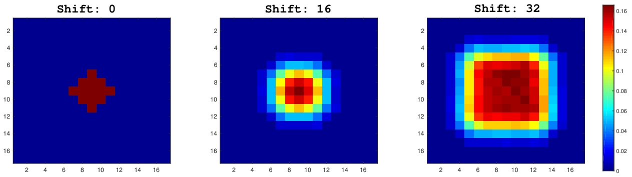


Figure 1. Visualization of prior probabilities of positive samples when using different random translations. The distributions become more uniform after random translations within ±32 pixels.

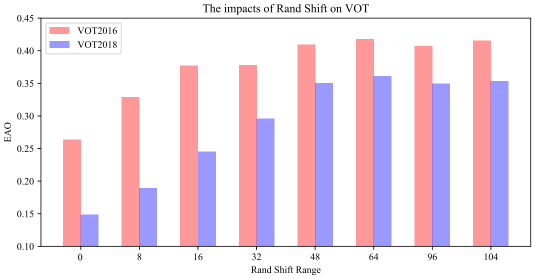


Figure 2. The impacts of the random translation on VOT dataset.

Strict translation invariance only exists in no padding network such as modified AlexNet [1]. Previous Siamese based Networks [1, 44, 43, 25, 54] are designed to be shallow to satisfy this restriction. However, if the employed networks are replaced by modern networks like ResNet or MobileNet, padding is inevitable to make the network going deeper, which destroys the strict translation invariance restriction. Our hypothesis is that the violation of this restriction will lead to a spatial bias.

We test our hypothesis by simulation experiments on a network with padding. Shift is defined as the max range of translation generated by a uniform distribution in data augmentation. Our simulation experiments are performed as follows. First, targets are placed in the center with different shift ranges (0, 16 and 32) in three sepreate training experiments. After convergence, we aggregate the heatmaps generated on test dataset and then visualize the results in Fig. 1. In the first simulation with zero shift, the probabilities on the border area are degraded to zero. It shows that a strong center bias is learned despite of the appearances of test targets. The other two simulations show that increasing shift ranges will gradually prevent model collapse into this trivial solution. The quantitative results illustrate that the aggregated heatmap of 32-shift is closer to the location distribution of test objects. It proves that this sampling strategy effectively alleviate the break of strict translation invariance property caused by the networks with padding.

To avoid putting a strong center bias on objects, we train SiamRPN with a ResNet-50 backbone by the *spatial aware sampling strategy* via sampling the target by a uniform distribution on the search image. As shown in Fig. 2, the performance with zero shift reduced to 0.14 on VOT2018, a suitable shift (±64 pixels) is vital for training a deep Siamese tracker.

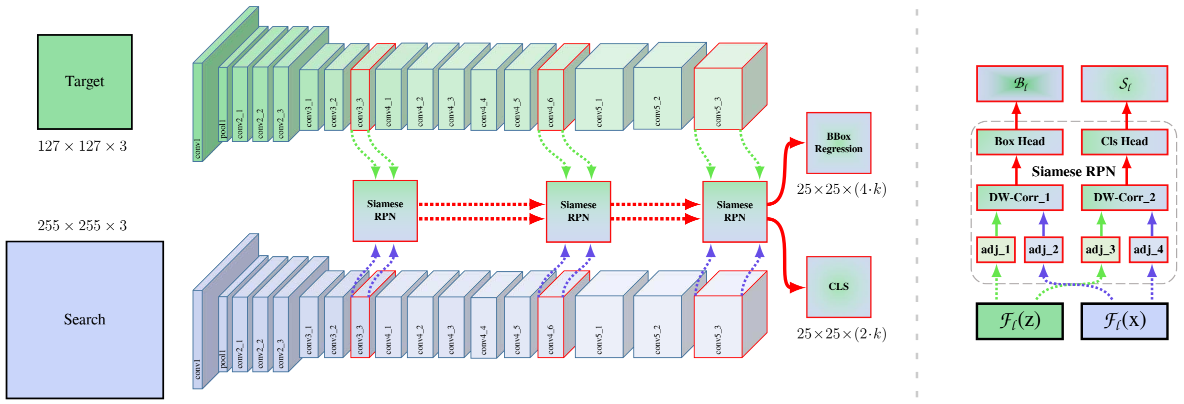


Figure 3. Illustration of our proposed framework. Given a target template and search region, the network ouputs a dense prediction by fusion the outputs from multiple Siamese Region Proposal (SiamRPN) blocks. Each SiamRPN block is shown on right.

3.2. ResNet-driven Siamese Tracking

Based on the above analyses, the influence of center bias can be eliminated. Once we eliminate the learning bias to the center location, any off-the-shelf networks (*e.g.*, MobileNet, ResNet) can be utilized to perform visual tracking after transfer learning. Moreover, we can adaptively construct the network topology and unveil the performance of *deep* network for visual tracking.

In this subsection, we will discuss how to transfer a deep network into our tracking algorithms. In particular, we conduct our experiments mainly focusing on ResNet-50 [15]. The original ResNet has a large stride of 32 pixels, which is not suitable for dense Siamese network prediction. As shown in Fig.3, we reduce the effective strides at the last two block from 16 pixels and 32 pixels to 8 pixels by modifying the $conv4$ and $conv5$ block to have unit spatial stride, and also increase its receptive field by dilated convolutions [28]. An extra 1×1 convolution layer is appended to each of block outputs to reduce the channel to 256.

Since the paddings of all layers are kept, the spatial size of the template feature increases to 15, which imposes a heavy computational burden on the correlation module. Thus we crop the center 7×7 regions [43] as the template feature where each feature cell can still capture the entire target region.

Following [25], we use a combination of cross correlation layers and fully convolutional layers to assemble a head module for calculating classification scores (denoted by $\mathcal{S}$) and bounding box regressor (denoted by $\mathcal{B}$). The Siamese RPN blocks are denoted by $\mathcal{P}$.

Furthermore, we find that carefully fine-tuning ResNet will boost the performance. By setting learning rate of ResNet extractor with 10 times smaller than RPN parts, the feature representation can be more suitable for tracking tasks. Different from traditional Siamese approaches, the parameters of the deep network are jointly trained in an end-to-end fashion. To the best of our knowledge, we are the first to achieve an end-to-end learning on a deep Siamese Network (> 20 layers) for visual tracking.

3.3. Layer-wise Aggregation

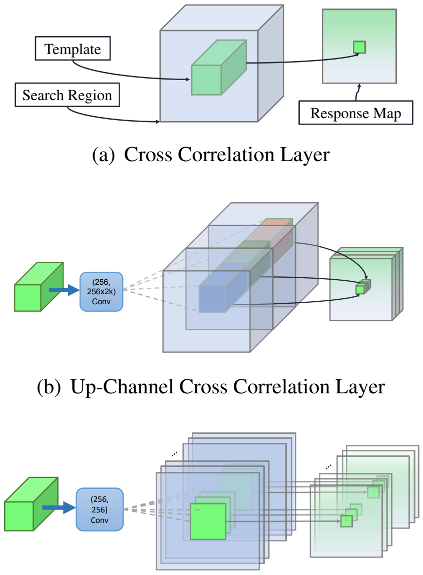
After utilizing deep network like ResNet-50, aggregating different deep layers becomes possible. Intuitively, visual tracking requires rich representations that span levels from low to high, scales from small to large, and resolutions from fine to coarse. Even with the depth of features in a convolutional network, a layer in isolation is not enough: compounding and aggregating these representations improve inference of recognition and localization.

In the previous works which only use shallow networks like AlexNet, multi-level features cannot provide very different representations. However, different layers in ResNet are much more meaningful considering that the receptive field varies a lot. Features from earlier layers will mainly focus on low level information such as color, shape, are essential for localization, while lacking of semantic information; Features from latter layers have rich semantic information that can be beneficial during some challenge scenarios like motion blur, huge deformation. The use of this rich hierarchical information is hypothesized to help tracking.

In our network, multi-branch features are extracted to collaboratively infer the target localization. As for ResNet-50, we explore multi-level features extracted from the last three residual block for our layer-wise aggregation. We refer these outputs as $\mathcal{F}_3(\mathbf{z})$, $\mathcal{F}_4(\mathbf{z})$, and $\mathcal{F}_5(\mathbf{z})$, respectively. As shown in Fig. 3, the outputs of $conv3$, $conv4$, $conv5$ are fed into three Siamese RPN module individually.

Since the output sizes of the three RPN modules have the same spatial resolution, weighted sum is adopted directly on the RPN output. A weighted-fusion layer combines all the outputs.

$$\mathcal{S}_{all} = \sum_{l=3}^{5} \alpha_i * \mathcal{S}_l, \quad \mathcal{B}_{all} = \sum_{l=3}^{5} \beta_i * \mathcal{B}_l. \tag{2}$$



(a) Cross Correlation Layer

(b) Up-Channel Cross Correlation Layer

(c) Depth-wise Cross Correlation Layer

Figure 4. Illustrations of different cross correlation layers. (a) Cross Correlation (XCorr) layer predicts a single channel similarity map between target template and search patches in SiamFC [1]. (b) Up-Channel Cross Correlation (UP-XCorr) layer outputs a multi-channel correlation features by cascading a heavy convolutional layer with several independent XCorr layers in SiamRPN [25]. (c) Depth-wise Cross Correlation (DW-XCorr) layer predicts multi-channel correlation features between a template and search patches.

The combination weights are separated for classification and regression since their domains are different. The weight is end-to-end optimized offline together with the network.

In contrast to previous works, our approach does not explicitly combine convolutional features, but learn classifiers and regressions separately. Note that with the depth of the backbone network significantly increased, we can achieve substantial gains from the sufficient diversity of visual-semantic hierarchies.

3.4. Depthwise Cross Correlation

The cross correlation module is the core operation to embed two branches information. SiamFC [1] utilizes a Cross-Correlation layer to obtain a single channel response map for target localization. In SiamRPN [25], Cross-Correlation is extended to embed much higher level information such as anchors, by adding a huge convolutional layer to scale the channels (UP-Xcorr). The heavy up-channel module makes seriously imbalance of parameter distribution (*i.e.* the RPN module contains 20M parameters while the feature extractor only contains 4M parameters in [25]), which makes the training optimization hard in SiamRPN.

In this subsection, we present a lightweight cross correlation layer, named Depthwise Cross Correlation (DW-XCorr), to achieve efficient information association. The DW-XCorr layer contains 10 times fewer parameters than the UP-XCorr used in SiamRPN while the performance is on par with it.

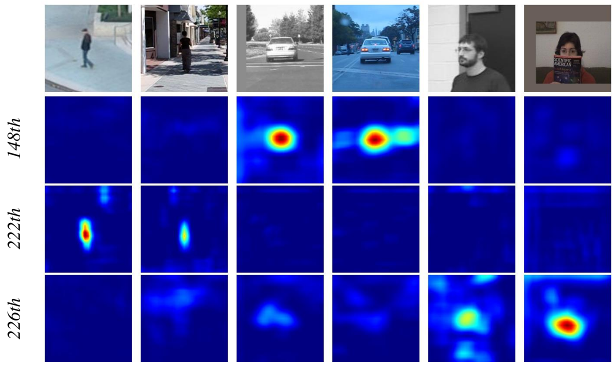


Figure 5. Channels of depthwise correlation output in *conv4*. There are totally 256 channels in *conv4*, however, only few of them have high response during tracking. Therefore we choose $148th$, $222th$, $226th$ channels as demonstration, which are $2nd$, $3rd$, $4th$ rows in the figure. The first row contains six corresponding search regions from OTB dataset [48]. Different channels represent different semantics, the $148th$ channel has high response on cars, while has low response on persons and faces. The $222th$ and $226th$ channel have high response on persons and faces, respectively.

To achieve this, a conv-bn block is adopted to adjust features from each residual blocks to suit tracking task. Crucially, the bounding box prediction and anchor based classification both are *asymmetrical*, which is different from SiamFC (See Sect. 3.1). In order to encode the difference, the template branch and search branch pass two *non-shared* convolutional layers. Then two feature maps with the same number of channels do the correlation operation channel by channel. Another conv-bn-relu block is appended to fuse different channel outputs. Finally, the last convolution layer for the output of classification or regression is appended.

By replacing cross-correlation to depthwise correlation, we can greatly reduce the computational cost and the memory usage. In this way, the numbers of parameters on the template and the search branches are balanced, resulting the training procedure more stable.

Furthermore, an interesting phenomena is illustrated in Fig.5. The objects in the same category have high response on same channels (car in $148th$ channel, person in $222th$ channel, and face in $226th$ channel), while responses of the rest channels are suppressed. This property can be comprehended as the channel-wise features produced by the depthwise cross correlation are nearly orthogonal and each channel represents some semantic information. We also analyze the heatmaps when using the up-channel cross correlation and the reponse maps are less interpretable.

4. Experimental Results

4.1. Training Dataset and Evaluation

Training. The backbone network of our architecture [15] is pre-trained on ImageNet [37] for image labeling, which has proven to be a very good initialization to other tasks [14, 28]. We train the network on the training sets of COCO [26], ImageNet DET [37], ImageNet VID, and YouTube-BoundingBoxes Dataset [36] and to learn a generic notion of how to measure the similarities between general objects for visual tracking. In both training and testing, we use single scale images with 127 pixels for template patches and 255 pixels for searching regions.

Evaluation. We focus on the short-term single object tracking on OTB2015 [48], VOT2018 [22] and UAV123 [32]. We use VOT2018-LT [22] to evaulate the long-term setting. In the long-term tracking, the object may leave the field of view or become fully occluded for a long period, which are more challenging than short-term tracking. We also analyze the generalization of our method on LaSOT [10] and TrackingNet [31], two of the recent largest benchmarks for single object tracking.

4.2. Implementation Details

Network Architecture. In experiments, we follow [54] for the training and inference settings. We attach two sibling convolutional layers to the stride-reduced ResNet-50 (Sect. 3.2) to perform proposal classification and bounding box regression with 5 anchors. Three randomly initialized 1×1 convolutional layers are attached to *conv3*, *conv4*, *conv5* for reducing the feature dimension to 256.

Optimization. SiamRPN++ is trained with stochastic gradient descent (SGD). We use synchronized SGD over 8 GPUs with a total of 128 pairs per minibatch (16 pairs per GPU), which takes 12 hours to converge. We use a warmup learning rate of 0.001 for first 5 epoches to train the RPN braches. For the last 15 epoches, the whole network is end-to-end trained with learning rate exponentially decayed from 0.005 to 0.0005. Weight decay of 0.0005 and momentum of 0.9 are used. The training loss is the sum of classification loss and the standard smooth L_1 loss for regression.

4.3. Ablation Experiments

Backbone Architecture. The choice of feature extractor is crucial as the number of parameters and types of layers directly affect memory, speed, and performance of the tracker. We compare different network architectures for the visual tracking. Fig. 6 shows the performance of using AlexNet, ResNet-18, ResNet-34, ResNet-50, and MobileNet-v2 as backbones. We report performance by Area Under Curve (AUC) of success plot on OTB2015 with respect to the top1 accuracy on ImageNet. We observe that our SiamRPN++ can benefit from *deeper* ConvNets.

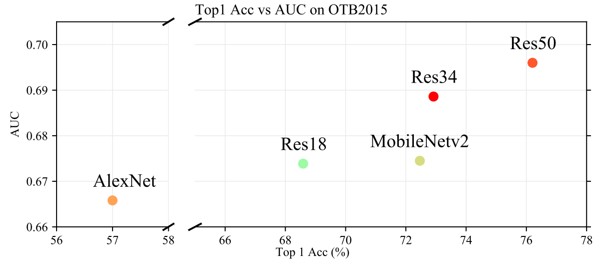


Figure 6. The Top-1 accuracy on ImageNet *vs.* AUC score on OTB2015.

BackBone	L3	L4	L5	Finetune	Corr	VOT2018	OTB2015
AlexNet					UP	0.332	0.658
					DW	0.355	0.666
ResNet-50	✓	✓	✓		UP	0.371	0.664
	✓	✓	✓	✓	UP	0.390	0.684
ResNet-50	✓			✓	DW	0.331	0.669
		✓		✓	DW	0.374	0.678
			✓	✓	DW	0.320	0.646
	✓	✓		✓	DW	0.346	0.677
	✓		✓	✓	DW	0.336	0.674
		✓	✓	✓	DW	0.383	0.683
ResNet-50	✓	✓	✓		DW	0.395	0.673
	✓	✓	✓	✓	DW	**0.414**	**0.696**

Table 1. Ablation study of the proposed tracker on VOT2018 and OTB2015. L3, L4, L5 represent *conv3*,*conv4*,*conv5*, respectively. Finetune represents whether the backbone is trained offline. Up/DW means Up channel correlation and depthwise correlation.

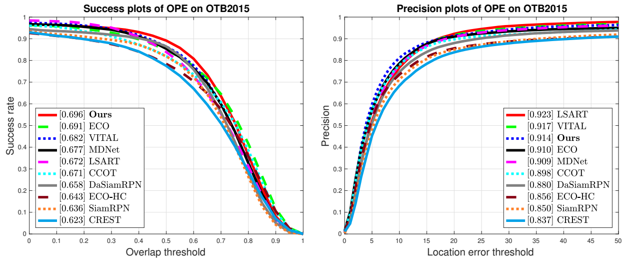
Table 1 also illustrates that by replacing AlexNet to ResNet-50, the performance improves a lot on VOT2018 dataset. Besides, our experiments shows that finetuning the backbone part is critical, which yields a great improvement on tracking performance.

Layer-wise Feature Aggregation. To investigate the impact of layer-wise feature aggregation, first we train three variants with single RPN on ResNet-50. We empirically found that $conv4$ alone can achieve a competitive performance with 0.374 in EAO, while deeper layer and shallower layer perform with 4% drops. Through combining two branches, $conv4$ and $conv5$ gains improvement, however no improvement is observed on the other two combinations. Even though, the robustness has increased 10%, which is the key vulnerability of our tracker. It means that our tracker still has room for improvement. After aggregating all three layers, both accuracy and robustness steadily improve, with gains between 3.1% and 1.3% for VOT and OTB. In total, layer-wise feature aggregation yields a 0.414 EAO score on VOT2018, which is 4.0% higher than that of the single layer baseline.

Depthwise Correlation. We compare the original Up-Channel Cross Correlation layer with the proposed Depthwise Cross Correlation layer. As shown in the Table 1, the proposed depthwise correlation gains 2.3% improvement on VOT2018 and 0.8% improvement on OTB2015, which

	DLSTpp	DaSiamRPN	SA_Siam_R	CPT	DeepSTRCF	DRT	RCO	UPDT	SiamRPN	MFT	LADCF	**Ours**
EAO ↑	0.325	0.326	0.337	0.339	0.345	0.356	0.376	0.378	0.383	0.385	0.389	**0.414**
Accuracy ↑	0.543	0.569	0.566	0.506	0.523	0.519	0.507	0.536	0.586	0.505	0.503	**0.600**
Robustness ↓	0.224	0.337	0.258	0.239	0.215	0.201	0.155	0.184	0.276	**0.140**	0.159	0.234
AO ↑	0.495	0.398	0.429	0.379	0.436	0.426	0.384	0.454	0.472	0.393	0.421	**0.498**

Table 2. Comparison with the state-of-the-art in terms of expected average overlap (EAO), robustness, and accuracy on the VOT2018.



(a) Success Plot (b) Precision Plot

Figure 7. Success and precision plots show a comparison of our tracker with state-of-the-art trackers on the OTB2015 dataset.

demonstrates the importance of depthwise correlation. This is partly beacause a balanced parameter distribution of the two branches makes the learning process more stable, and converges better.

4.4. Comparison with the state-of-the-art

OTB-2015 Dataset. The standardized OTB benchmark [48] provides a fair testbed on robustness. The Siamese based tracker formulate the tracking as one-shot detection task without any online update, thus resulting in inferior performance on this no-reset setting benchmark. However, we identify the limited representation from the *shallow* network as the primary obstacle preventing Siamese based tracker from surpassing top-performing methods, such as C-COT variants [9, 5].

We compare our SiamRPN++ tracker on the OTB2015 with the state-of-the-art trackers. Fig. 7 shows that our SiamRPN++ tracker produces leading result in overlap success. Compared with the recent DaSiamRPN [54], our SiamRPN++ improves 3.8% in overlap and 3.4% in precision from the considerably increased depth. Representations extracted from deep ConvNets are less sensitive to illumination and background clutter. And to the best of our knowledge, this is the first time that Siamese tracker can obtain the comparable performance with the state-of-the-art tracker on OTB2015 dataset.

VOT2018 Dataset. We test our SiamRPN++ tracker on the lastest VOT-2018 dataset [22] in comparison with 10 state-of-the-art methods. The VOT-2018 public dataset is one of the most recent datasets for evaluating online model-free single object trackers, and includes 60 public sequences with different challenging factors. Following the evaluation protocol of VOT-2018, we adopt the Expected Average Overlap (EAO), Accuracy(A) and Robustness(R) and no-reset-based Average Overlap(AO) to compare different trackers. The detailed comparisons are reported in Table 2.

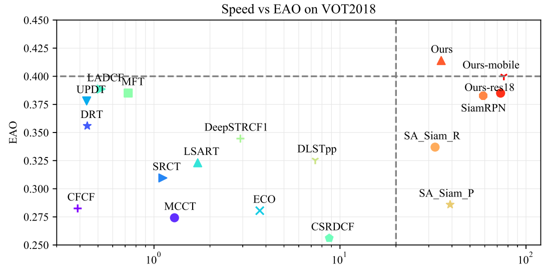


Figure 8. A comparison of the quality and the speed of state-of-the-art tracking methods on VOT2018. We visualize the Expected Average Overlap (EAO) with respect to the Frames-Per-Seconds (FPS). Note that the FPS axis is in the log scale. Two of our variants, which replace ResNet-50 backbone with ResNet-18 (Ours-res18) and MobileNetv2 (Ours-mobile), respectively.

From Table 2, we observe that the proposed SiamRPN++ method achieves the top-ranked performance on EAO, A and AO criteria. Especially, our SiamRPN++ tracker outperforms all existing trackers, including the VOT2018 challenge winner. Compared with the best tracker in the VOT2018 challenge (LADCF [22]), the proposed method achieves a performance gain of 2.5%. In addition, our tracker achieves a substantial improvement over the challenge winner (MFT [22]), with a gain of 9.5% in accuracy.

In comparison with the baseline tracker DaSiamRPN, our approach yields substantial gains of 10.3% on robustness, which is the common vulnerability of the Siamese Network based tracker against correlation filters method. Even though, due to the lack of adaptation to the template, the robustness still has a gap with the state-of-art correlation filters methods [2] which relies on the online updating.

The One Pass Evaluation (OPE) is also adopted to evaluate trackers and the AO values are reported to demonstrate their performance. From the last row in Table 2, we can observe that our method achieves comparable performance compared to the DLSTpp [22] and improves the DaSiamRPN [54] method by an absolute gain of 10.0%.

Accuracy vs. Speed. In Fig. 8, we visualize the EAO on VOT2018 with respect to the Frames-Per-Second (FPS). The reported speed is evaluated on a machine with an NVIDIA Titan Xp GPU, other results are provided by the VOT2018 official results. From the plot, our SiamRPN++ achieves best performance, while still running at realtime speed(35 FPS). It is worth noting that two of our variants achieve nearly the same accuracy as SiamRPN++, while running at more than 70 FPS, which makes these two variants highly competitive.

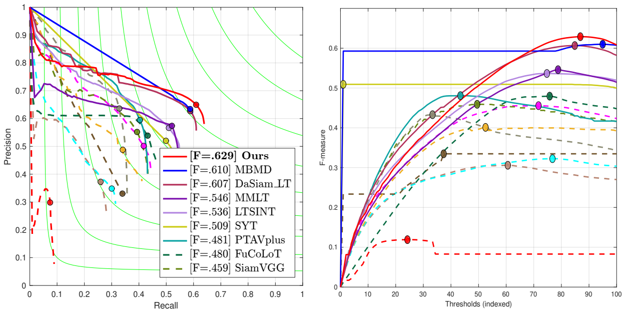


Figure 9. Long-term tracking performance. The average tracking precision-recall curves (left), the corresponding F-score curves (right). Tracker labels are sorted according to the F-score.

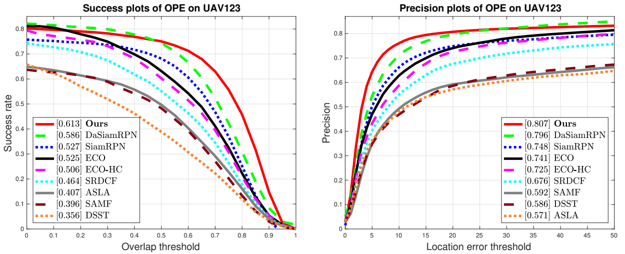


Figure 10. Evaluation results of trackers on UAV123.

VOT2018 Long-term Dataset. In the latest VOT2018 challenge, a long-term experiment are newly introduced. It is composed of 35 long sequences, where targets may leave the field of view or become fully occluded for a long period. The performance measures are precision, recall and a combined F-score. We report all these metrics compared with the state-of-the-art trackers [41, 11] on VOT2018-LT.

As shown in the Fig. 9, after equipping our tracker with the long term strategy, SiamRPN++ obtains 2.2% gain from DaSiam_LT, and outperforms the best tracker by 1.9% in F-score. The powerful feature extracted by ResNet improves both TP and TR by 2% absolutely from our baseline DaSiamRPN. Meanwhile, the long term version of SiamRPN++ is still able to run at 21 FPS, which is nearly 8 times faster than MBMD [22], the winner of VOT2018-LT.

UAV123 Dataset. UAV123 dataset includes 123 sequences with average sequence length of 915 frames. Besides the recent trackers in [30], ECO [5], ECO-HC [5], DaSiamRPN [54], SiamRPN [25] are added on comparison. Fig. 10 illustrates the precision and success plots of the compared trackers. Specifically, our tracker achieves a success score of 0.613, which outperforms DaSiamRPN (0.586) and ECO (0.525) with a large margin.

LaSOT Dataset. To further validate the proposed framework on a larger and more challenging dataset, we conduct experiments on LaSOT [10]. The LaSOT dataset provides a large-scale, high-quality dense annotations with 1,400 videos in total and 280 videos in the testing set. Fig. 11 reports the overall performances of our SiamRPN++ tracker on LaSOT testing set. Without bells and whistles, our SiamRPN++ model is sufficient to achieve state-of-the-art AUC score of 49.6%. Specifically, SiamRPN++ increases the normalized distance precision and AUC relatively by 23.7% and 24.9% over MDNet [33], which is the best tracker reported in the original paper.

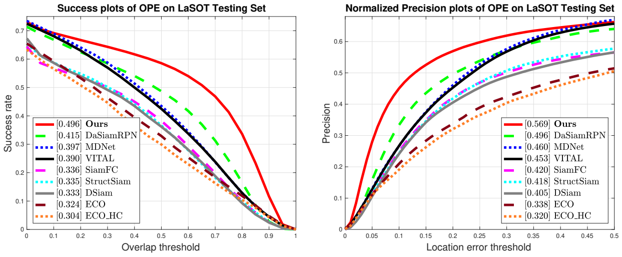


Figure 11. Evaluation results of trackers on LaSOT.

	CSRDCF [29]	ECO [5]	SiamFC [1]	CFNet [43]	MDNet [33]	DaSiamRPN [54]	**Ours**
AUC (%)	53.4	55.4	57.1	57.8	60.6	63.8	**73.3**
P (%)	48.0	49.2	53.3	53.3	56.5	59.1	**69.4**
P_{norm} (%)	62.2	61.8	66.3	65.4	70.5	73.3	**80.0**

Table 3. State-of-the-art comparison on the TrackingNet `test` set in terms of success, precision, and normalized precision.

TrackingNet Dataset. The recently released TrackingNet [31] provides a large amount of data to assess trackers in the wild. We evaluate SiamRPN++ on its `test` set with 511 videos. Following [31], we use three metrics success (AUC), precision (P) and normalized precision (P_{norm}) for evaluation. Table 3 demonstrates the comparison results to trackers with top AUC scores, showing that SiamRPN++ achieves the best results on all three metrics. In specific, SiamRPN++ obtains the AUC score of 73.3%, P score of 69.4% and P_{norm} score of 80.0%, outperforming the second best tracker DaSiamRPN [54] with AUC score of 63.8%, P score of 59.1% and P_{norm} score of 73.4% by 9.5%, 10.3% and 6.6%, respectively.

In summary, it is important to note that all these consistent results show the generalization ability of SiamRPN++.

5. Conclusions

In this paper, we have presented a unified framework, referred as SiamRPN++, to end-to-end train a deep Siamese network for visual tracking. We show theoretical and empirical evidence that how to train a deep network on Siamese tracker. Our network is composed of a multi-layer aggregation module which assembles the hierarchy of connections to aggregate different levels of representation and a depth-wise correlation layer which allows our network to reduce computation cost and redundant parameters while also leading to better convergence. Using SiamRPN++, we obtained state-of-the-art results on the VOT2018 in real-time, showing the effectiveness of SiamRPN++. SiamRPN++ also acheived state-of-the-art results on large datasets like LaSOT and TrackingNet showing its generalizability.

References

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016. 1, 2, 3, 5, 8

[2] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg. Unveiling the power of deep tracking. In *ECCV*, September 2018. 7

[3] D. Bolme, J. Beveridge, B. Draper, and Y. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 2

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2

[5] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 1, 2, 7, 8

[6] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015. 2

[7] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCV Workshops*, 2015. 2

[8] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. De Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014. 2

[9] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016. 2, 7

[10] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling. Lasot: A high-quality benchmark for large-scale single object tracking, 2018. 2, 6, 8

[11] H. Fan and H. Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, 2017. 8

[12] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, 2017. 1

[13] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, 2017. 2

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 6

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 4, 6

[16] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016. 1, 2

[17] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 2015. 2

[18] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *CVPR*, 2015. 2

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[20] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, and L. Cehovin Zajc. The visual object tracking vot2016 challenge results. In *ECCV Workshops*, 2015. 2

[21] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, and L. Cehovin Zajc. The visual object tracking vot2017 challenge results. In *ICCV*, 2017. 2

[22] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pfugfelder, L. C. Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, G. Fernandez, and et al. The sixth visual object tracking vot2018 challenge results. In *ECCV Workshops*, 2018. 2, 6, 7, 8

[23] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. ˇ Cehovin, and G. Fern´. The visual object tracking vot2015 challenge results. In *ICCV Workshops*, 2015. 2

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[25] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 1, 2, 3, 4, 5, 8

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6

[27] L. Liu, J. Xing, H. Ai, and X. Ruan. Hand posture recognition using finger geometric feature. In *ICIP*, 2012. 1

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 4, 6

[29] A. Lukezic, T. Vojir, L. Cehovin Zajc, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, 2017. 8

[30] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, pages 445–461. Springer, 2016. 8

[31] M. Müller, A. Bibi, S. Giancola, S. Al-Subaihi, and B. Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. *ECCV*, 2018. 2, 6, 8

[32] M. Müller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016. 2, 6

[33] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 2, 8

[34] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. In *CVPR*, 2018. 2

[35] R. Pflugfelder. An in-depth analysis of visual tracking with siamese neural networks. *arXiv:1707.00569*, 2017. 3

[36] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 7464–7473. IEEE, 2017. 6

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 6

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[40] W. Tang, P. Yu, and Y. Wu. Deeply learned compositional models for human pose estimation. In *ECCV*, 2018. 3

[41] R. Tao, E. Gavves, and A. W. Smeulders. Tracking for half an hour. *arXiv preprint arXiv:1711.10217*, 2017. 8

[42] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016. 1, 2, 3

[43] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017. 1, 2, 3, 4, 8

[44] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu. Dcfnet: Discriminant correlation filters network for visual tracking. In *arXiv:1704.04057*, 2017. 1, 2, 3

[45] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank. Learning attentions: Residual attentional siamese network for high performance online visual tracking. In *CVPR*, 2018. 1, 2

[46] Q. Wang, M. Zhang, J. Xing, J. Gao, W. Hu, and S. Maybank. Do not lose the details: Reinforced representation learning for high performance visual tracking. In *IJCAI*, 2018. 1, 2

[47] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 2

[48] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *TPAMI*, 2015. 1, 2, 5, 6, 7

[49] J. Xing, H. Ai, and S. Lao. Multiple human tracking based on multi-view upper-body detection and discriminative learning. In *ICPR*, 2010. 1

[50] G. Zhang and P. Vela. Good features to track for visual slam. In *CVPR*, 2015. 1

[51] M. Zhang, Q. Wang, J. Xing, J. Gao, P. Peng, W. Hu, and S. Maybank. Visual tracking via spatially aligned correlation filters network. In *ECCV*, 2016. 2

[52] M. Zhang, J. Xing, J. Gao, and W. Hu. Robust visual tracking using joint scale-spatial correlation filters. In *ICIP*, 2015. 2

[53] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu. Joint scale-spatial correlation tracking with adaptive rotation estimation. In *ICCV Workshops*, 2015. 2

[54] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018. 1, 2, 3, 6, 7, 8