SiamPolarMask: Unifying Object Tracking and Segmentation

Puhua Jiang*, Weichen Zhang*, Jinxin Zhou*, Zongyu Li*
Department of Electrical Engineering and Computer Science
University of Michigan
zonyul@umich.edu

Abstract

By decomposing the object tracking task into two subproblems as classification for pixel category and instance segmentation for the object by polar representation, we propose a novel network called SiamPolarMask, which takes advantages of two different models, SiamCar and Polar Mask. SiamPolarMask consists of feature extraction subnetwork, which takes ResNet-50 as backbone, and one classification-mask regression subnetwork for the instance segmentation.

Firstly we train our model on the MS COCO dataset to see if our model can handle the object detection tasks on a single image, next we finetune our model on DAVIS dataset, in order to let our model learn how the information flows between frames in a video. Then we show that our model can successfully handle the object detection and segmentation task by outputting precise masks for the object in a single image. Next we show our model can successfully track the object for some videos but fails for some other videos. Finally we give conclusions of our model and discuss about the future work.

1. Introduction

Visual object tracking has received considerable attention due to its wide application such as intelligent surveillance, human-machine interaction and autonomous driving vehicles. However, it remains a challenging task, since the object in unconstrained recording conditions often suffers from large illumination variation, scale variation, background clutters and heavy occlusions, etc.

Researchers have proposed several tracking methods [1, 3, 4, 8, 10, 13], which are all around the Siamese network architectures. The Siamese network formulates the visual tracking task as a target matching problem and aims to learn a general similarity gap between the target template and the search region. SiamRPN [3] attaches the Siamese network a region proposal extraction subnetwork (RPN). By jointly training a classification branch and a regression branch for visual tracking, SiamRPN avoids the time-consuming step of extracting multi-scale feature maps for the object scale invariance. It achieves state-of-the-art results on many benchmarks. Later works such as DaSiam [22], CSiam [8] and SiamRPN++ [4] improves SiamRPN. However, since anchors are introduced for region proposal, these trackers are sensitive to the numbers, sizes and aspect ratios of anchor boxes, and expertise on hyper-parameter tuning is crucial to obtain successful tracking with these trackers.

SiamCar [5] is a simple yet efficient Siamese based classification and regression network to learn the classification and regression models simultaneously in an end-to-end manner. One significant improvement in SiamCar is that it is both anchor and proposal free, which means the number of hyper-parameters would be significantly reduced, so it prevents SiamCar from complicated parameter tuning and makes the SiamCar much simpler and easier, especially in training. However, one limit to the SiamCar is it can't do the instance segmentation. SiamMask [16] is a unifying approach to do the classification, bounding box regression and instance segmentation at the same time, by adding a new branch into the Siamese network to predict a binary segmentation mask. This network is powerful but there is a simpler way to achieve a similar performance without adding more branches.

PolarMask [6] is an instance segmentation method based on the Polar Representation since its inherent advantages are as follows:

(1) The origin point of the polar coordinate can be seen as the center of object.

(2) Starting from the origin point, the point in contour is determined by the distance and angle.

(3) The angle is naturally directional and makes it very convenient to connect the points into a whole contour.

(4) Furthermore, PolarMask can be viewed as a generalization of FCOS [21]. In other words, FCOS is a special case of PolarMask since bounding boxes can be viewed as the simplest mask with only 4 directions.

In order to maximize the advantages of Polar Representation, we use Polar Centerness and Polar IoU loss with sampling high-quality center examples and optimization for dense distance regression, respectively. The main contributions of this work are:

(1) We combine the advantages of SiamCar and PolarMask network, and propose a new framework called SiamPolarMask, which predicts a mask in polar coordinate instead of the usual binary segmentation mask without adding new branches to the framework.

(2) We show that our model, SiamPolarMask, can successfully handle the object detection task and the segmentation task for a single image on MS COCO [20] dataset and DAVIS [7] dataset.

(3) We also show that our proposed model can have a competitive tracking performance on DAVIS dataset, but less accurate object segmentation performance in the successive frames, especially the last several frames.

The remainder of the paper is organized as follows. **Section 2** describes the related works in tracking based on Siamse network. **Section 3** illustrates our methods on visual tracking and instance segmentation for this project. **Section 4** summarizes the implementation details on data preparation, training and testing. **Section 5** lists some results of our neural network outputs, testing on the MS COCO [20] and DAVIS [7]. **Section 6** concludes our method and discuss about the future work.

2. Related Works

2.1. Object Visual Tracking

Tracking researchers devote to design faster and more accurate trackers from different aspects like feature extraction [9], template updating [10, 11], classifier design [14] and bounding box regression [2]. Early feature extraction mainly uses color features, texture features or other handcrafted ones. Benefit by the development of deep learning, now the deep convolutional feature CNN is widely adopted. Template updating can improve the model adaptability, but online tracking is very inefficient. Besides, the tracking drift problem for template updating still need to be solved.

As one of the pioneering works, SiamFC [13] constructs a fully convolutional Siamese network to train a tracker. Encouraged by its success, many researchers follow the work and propose some updated models [1, 3, 4, 10]. CFNet [10] introduces the Correlation Filter layer to the SiamFC framework and performs online tracking to improve the accuracy. By modifying the Siamese branches with two online transforms, DSiam [15] proposes to learn a dynamic Siamese network, which achieves an improved accuracy with acceptable speed loss. The SAsiam [1] builds a twofold Siamese network with a semantic branch and an appearance branch. The two branches are trained separately to keep the heterogeneity of features but combined at the testing time to improve the tracking accuracy. In order to deal with the scale variation problem, these Siamese networks need to process multi-scale searching and result in time-consuming problem.

Inspired by the region proposal network for object detection [18], the SiamRPN [3] tracker performs the region proposal extraction after the Siamese network outputs. By jointly training a classification branch and a regression branch for region proposal, SiamRPN avoids the time-consuming step of extracting multi-scale feature maps for the object scale invariance and achieves very efficient results. However, it has difficulty in deal with distractors with similar appearance to the object. Based on SiamRPN, DaSiamRPN [22] increases the hard negative training data during the training phase. Through data enhancement, they improve the discrimination of the tracker and obtain a much more robust result. The tracker is further extended to long term visual tracking. Up to now the framework has been modified a lot from SiamFC, but the performance still can not move on with deeper network by using AlexNet as backbone. Aiming to this problem, SiamRPN++ [4] optimizes the network architecture by using the ResNet [12] as backbone. At the same time, they randomly shift the training object location in the search region during model training to eliminate the center bias. After these modifications, the better tracking accuracy can be achieved in a very deep network architecture instead of shallow neural networks.

SiamCar is a framework that can greatly alleviate the problems that are:

(1) The tracking performance is very sensitive to the relative hyper-parameters of anchors, which need to be carefully tuned so empirical tricks are involved to achieve ideal performance.

(2) Moreover, since the size and aspect ratio of anchor boxes are fixed, even with heuristic tuned parameters, these trackers still have difficulty in processing objects with large shape deformation and pose variation.

2.2. Polar Mask

Polar representation was firstly used in [17] to detect cells in microscopic images, where the problem is much simpler as there are only two categories. Concurrent to the PolarMask is the work of ESESeg [19], which also employs the polar coordinate to model instances. However, PolarMask [6] achieves significantly better performance than ESESeg due to very different designs other than the polar representation. Note that most of these methods do not model instances directly and they can sometimes be hard to optimize (*e.g.* longer training time, more data augmentation

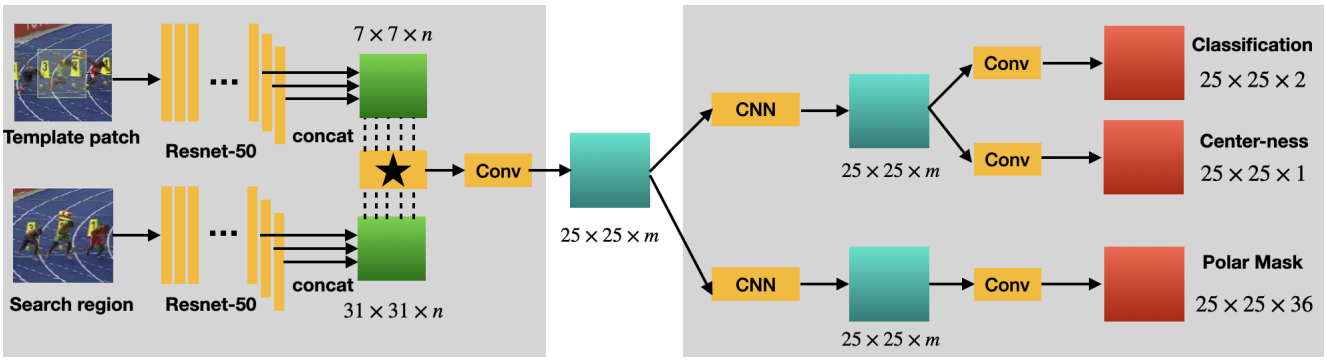

Figure 1. Architecture of our proposed neural network

and extra labels). PolarMask directly models instance segmentation with a much simpler and flexible way of two paralleled branches: classifying each pixel of mass-center of instance and regressing the dense distance of rays between mass-center and contours. The most significant advantage of PolarMask is being simple and efficient compared with the above methods.

3. Method

3.1. Network Architecture

We propose a neural network consisting of two parts: one Siamese subnetwork for feature extraction and one classification-regression subnetwork for polar mask prediction. We use a pretrained ResNet-50 as backbone, then freeze it during the training, so only the classification-regression subnetwork is trained. The network architecture is shown in Fig.1, where the $\star$ denotes the depth-wise cross correlation.

3.2. Polar Representation

Given an instance mask, first sample a candidate center (x_c, y_c) of the instance and the point located on the contour $(x_i, y_i), i = 1, 2, ..., N$. Then, starting from the center, n rays are emitted uniformly with the same angle interval $\Delta\theta$ (*e.g.*, $n = 36, \Delta\theta = 10^\circ$), whose length is determined from the center to the contour. In this way, the polar representation model the instance mask in the polar coordinate as one center and n rays. Since the angle interval is pre-defined, only the length of the ray needs to be predicted.

3.3. Polar Centerness

Centerness [21] is introduced to suppress the low quality detected objects without introducing any hyper parameters and it turns out to be effective in object bounding box detection.

Given a set $\{d_1, d_2, ..., d_n\}$ for the length of n rays of one instance, the polar centerness is given by:

$$\text{Polar Centerness} = \sqrt{\frac{\min(\{d_1, d_2, ..., d_n\})}{\max(\{d_1, d_2, ..., d_n\})}} \tag{1}$$

3.4. Polar IoU Loss

As shown in Fig.2, in the polar coordinate system, for one instance, mask IoU is calculated as follows:

$$\text{IoU} = \frac{\int_0^{2\pi} \frac{1}{2} \min(d, d^*)^2 d\theta}{\int_0^{2\pi} \frac{1}{2} \max(d, d^*)^2 d\theta} \tag{2}$$

where the regression target d and predicted d^* are length of the ray, and the angle is θ. Its discrete form can be expressed as:

$$\text{IoU} = \lim_{N \to \infty} \frac{\sum_{i=1}^{N} \frac{1}{2} d_{\min}^2 \Delta\theta_i}{\sum_{i=1}^{N} \frac{1}{2} d_{\max}^2 \Delta\theta_i} \tag{3}$$

If ignoring the power form, then a simplified IoU can be written as:

$$\text{Polar IoU} = \frac{\sum_{i=1}^{N} d_{\min}}{\sum_{i=1}^{N} d_{\max}} \tag{4}$$

Polar IoU Loss is the binary cross entropy (BCE) loss of Polar IoU. Since the optimal IoU is always 1, the loss is actually is negative logarithm of Polar IoU:

$$\text{Polar IoU Loss} = \log \frac{\sum_{i=1}^{N} d_{\max}}{\sum_{i=1}^{N} d_{\min}} \tag{5}$$

Therefore, the overall loss function is:

$$L = L_{cls} + \lambda_1 L_{cen} + \lambda_2 L_{reg} \tag{6}$$

where L_{cls} represents the cross-entropy loss for the classification. During model training, L_{cen} represents the binary cross entropy (BCE) loss for the polar centerness. We choose $\lambda_1 = 1$ and $\lambda_2 = 1.5$ empirically.

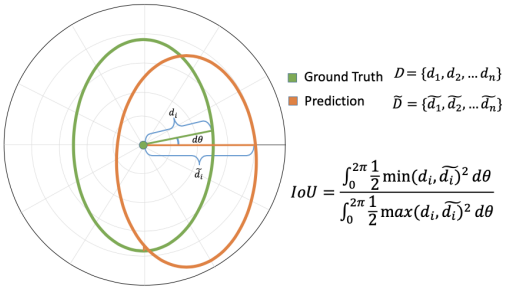

Figure 2. Mask IoU in Polar Representation

3.5. Tracking Phase

Tracking aims at predicting a mask for the target in the current frame. For a location (i, j), the proposed framework can produce a 38D vector $T_{ij} = (cls, cen, mask)$, where cls represents the foreground score of classification, cen represents the centerness score, mask represents the predicted mask of the target in the current frame. During tracking, the size and aspect ratio of the mask typically see minor change across consecutive frames. To supervise the prediction using this spatial temporal consistency, we adopt a scale change penalty p_{ij} as that introduced in [3] to re-rank the classification score cls, which admits an updated 4D vector $PT'_{ij} = (cls_{ij} \times p_{ij}, cen, width, height)$ where width and height are calculated from mask. Then the tracking phase can be formulated as:

$$q = (\hat{i}, \hat{j}) = \arg\max_{i,j} \{(1 - \lambda_d) cls_{ij} \times p_{ij} + \lambda_d H\} \quad (7)$$

where H is the cosine window and λ_d is the balance weight. The output q is a queried location with the highest score to be a target pixel.

3.6. RoI Mapping

This subsection discusses how to calculate the relationship of pixel points between the original image and the feature map. The idea is that we project the center of a window onto a pixel in the feature map, such that this point (in the original image domain) is the closest to the center of receptive field. For convolutional layer, given the stride and kernel size, the receptive field can be expressed as follows:

$$RF_{i-1} = (RF_i - 1) \times s + k \quad (8)$$

where RF_i is the top layer's receptive field at the ith layer, s is the stride and k is the kernel size. If taking padding into consideration, the relationship formulation becomes:

$$P_i = s_i \times P_{i+1} + (k_i - 1)/2 - padding \quad (9)$$

where P_i denotes the coordinates in ith layer feature map. In dilation case, we can still take it as the expansion of the kernel size. Pooling layer also follows the same relation. For activation layer like ReLU, since the receptive field doesn't change, there is no change in the coordinate.

4. Implementation

In our experiment, we take datasets from MS COCO [20] and DAVIS [7]. Due to the training time and machine capacity, we pick about 2k instances from MS COCO trainval2017 and 20 videos from DAVIS for training. The input sizes of template and search regions follow the set in [5], respectively to 127 pixels and 255 pixels.

4.1. Data Preprocessing

Before being fed to our model, the template image and search image have to be transformed in 127 pixels and 255 pixels respectively with same process. We first calculate the area and center of the bounding box. Then we crop a square centered on it with the scaled area and finally resize them to the corresponding size.

4.2. Training Details

During the training process, the batch size is set as 64 for coco and 4 for DAVIS by using stochastic gradient (SGD) with an initial learning rate 0.001. Our backbone is ResNet-50 pretrained on ImageNet. We freeze the parameters in Siamese subnetwork and only update cls-reg subnetwork. We first train our model on MS COCO for 100 epochs and another 100 epochs on DAVIS.

4.3. Testing Details

During the testing process, we take use of the offline tracking strategy. Only the object in the initial frame of a sequence is adopted as the template patch. Consequently, the target branch of the Siamese subnetwork can be precomputed and fixed during the whole tracking period. The search region in the current frame is adopted as the input of the search branch. In Fig. 3 we show a whole tracking process.

With the outputs of classification-regression subnetwork, a location q is queried through Equation (7). In order to achieve a more stable and smoother prediction between adjacent frames, a weighted average of regression mask corresponding to the previous frame's mask and q are computed as the final tracking result. After we find the q in the (25,25) size result, we have to locate it in the (255,255) size search patch and then in the original image. Plugging into equations (8) and (9), the mapping relation between the result map and search patch can be calculated by:

$$\text{MaskCenter}_{\text{search}} = \text{MaskCenter}_{\text{result}} \times 8 + 31 \quad (10)$$

The mapping relation between search patch and the original

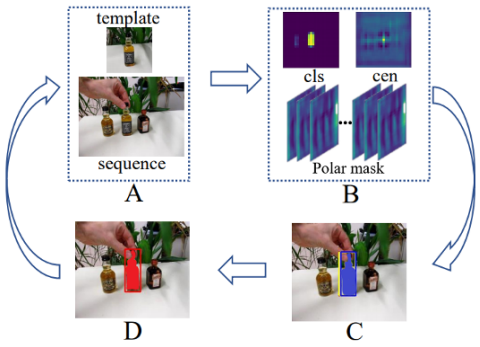

Figure 3. Tracking process: Sub-figure A shows a pair of inputs while Sub-figure B shows the corresponding outputs of the model. From Sub-figure B we can observe that the outputs of the model can depict good prediction for different attributes of the object. Sub-figure C shows the predicted bounding boxes and masks corresponding to this frame and last frame. Sub-figure D shows the final predicted bounding box and mask by averaging those boxes and masks in C.

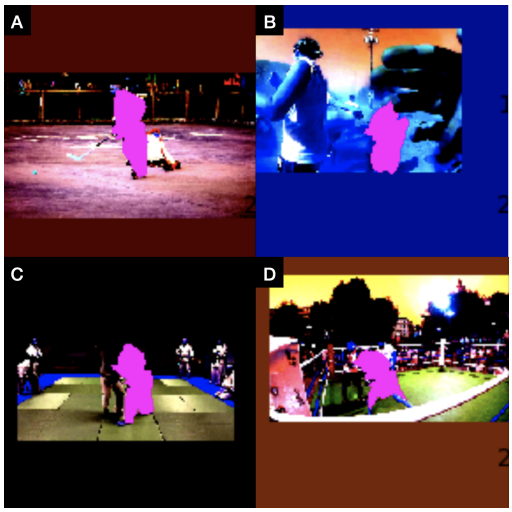

Figure 4. Detection and segmentation results on MS COCO. Sub-figure A shows a woman is dancing. Sub-figure B shows a boxer. Sub-figure C shows a man in judo competition. Sub-figure D shows a man in ice hockey.

image is :

$$\begin{aligned}\text{MaskCenter}_{\text{original}} = (\text{MaskCenter}_{\text{search}} - (127, 127)) \times s \\ + \text{BBoxCenter}_{\text{original}}\end{aligned} \tag{11}$$

where $\text{MaskCenter}_{\text{search}}$ is q and $(127, 127)$ is the bounding box center in search patch, s is the scale rate, $\text{BBoxCenter}_{\text{original}}$ is the bounding box center in original image (generated from last frame) and the $\text{MaskCenter}_{\text{original}}$ is the mask center in original image.

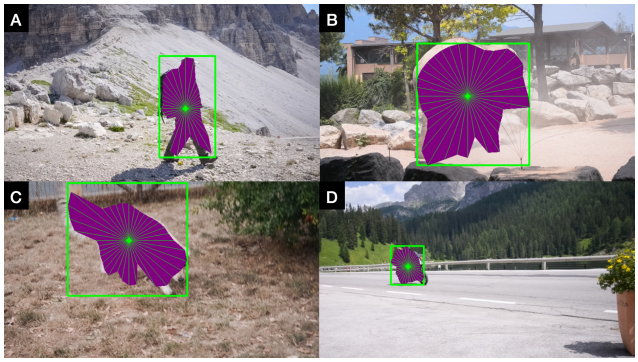

Figure 5. Detection and segmentation results on DAVIS. Sub-figure A shows a man going hiking, Sub-figure B shows an elephant moving along a rock road, Sub-figure C shows a dog walking on a lawn, Sub-figure D shows a motorcycle riding on the highway.

5. Result

5.1. Detection and Segmentation

We first test our model on MS COCO and DAVIS dataset, and let our CNN predict the bounding box as well as the polar mask just for single images. Fig. 4 and Fig. 5 show that our model has certain capabilities to capture the center of the objects and their contour masks, which means our model has the capibility to detect and segment object in a single image, as well as our cls-reg subnetworks are converged during the training.

5.2. Tracking

Next we test our model on the DAVIS dataset using the whole image sequence, we observe that the result is different. For some videos *e.g.* frames 1-5 at the top of Fig.6, where the object moves slowly, our tracker can successfully track the object by outputting a reasonable center and its corresponding mask. However, in some videos *e.g.* frames 1-5 at the bottom of Fig.6, where the object has a significant change of location, our tracker works only for the first few frames, then it fails to locate the object center and its corresponding mask also goes wrong and has a smaller and smaller length. We think this is because of the accumulation of errors, especially in the video where the object moves too quickly. On the other hand, it also shows that our model has not learned how the information flows between each frame in a video very well.

We spend a lot of time tuning our network and tracking algorithms to achieve reasonable results on this kind of videos, but they seem more challenging than we expect. Also due to the lack of time, we cannot present a comprehensive comparison between our model and other models in the literature.

Figure 6. Some frames outputs by our model. Frames 1-5 on the top shows a successful case of our model, Frames 1-5 on the bottom shows a failure tracking case of our model.

6. Conclusion

In this paper, we present a Siamese classification and segmentation framework called SiamPolarMask to end-to-end train a deep Siamese network for visual tracking. We show that the tracking task can be solved in a per-pixel manner and adopted with the neat fully convolutional framework.

Tracking and Segmentation are both challenging tasks, our model still have some problems as we talked above. Next step is to solve these problems, we will train our model on more datasets, train the backbone as well and adapt more tricks in tracking. We hope it will achieve state-of-the-art results on large dataset. Since the present framework is simple and neat, we hope it can be easily modified with specific modules to make further improvement in the future.

References

[1] A.F.He, C.Luo, X.M.Tian, and W.J.Wang. A two fold siamese network for real-time object tracking. *CVPR*, 2018. 1, 2

[2] B.Goutam. Accurate tracking by overlap maximization. 2019. 2

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

[4] B.Li, W.Wu, Q.Wang, F.Y.Zhang, J.L.Xing, and J.J.Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *CVPR*, 2019. 1, 2

[5] D.Guo, J.Wang, Y.Cui, Z.Wang, and S.Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. *CVPR*, 2020. 1, 4

[6] E.Xie, P.Sun, X.Song, W.Wang, D.Liang, C.Shen, and P.Luo. Polarmask: Single shot instance segmentation with polar representation. *CVPR*, 2020. 1, 2

[7] P. F, P.-T. J, and M. B. *et al*. A benchmark dataset and evaluation methodology for video object segmentation. *CVPR*, 2016. 2, 4

[8] H.Fan and H.B.Ling. Siamese cascaded region proposal networks for real-time visual tracking. *CVPR*, 2019. 1

[9] Joao.F.Henriques, R.Caseiro, M.Pedro, and B.Jorge. High-speed tracking with kernelized correlation filters. *TPAMI*, 2014. 2

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

[11] J.Y.Gao, T.Z.Zhang, and C.S.Xu. Graph convolutional tracking. *CVPR*, 2019. 2

[12] K.He, X.Zhang, S.Ren, and J.Sun. Deep residual learning for image recognition. *CVPR*, 2016. 2

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

[14] L.Zhang, V.Jagannadan, N.S.Ponnuthurai, A.Narendra, and M.Pierre. Robust visual tracking using oblique random forests. *CVPR*, 2019. 2

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

[16] Q.Wang, L.Zhang, L.Bertinetto, W.Hu, and P.Torr. Fast online object tracking and segmentation: A unifying approach. *CVPR*, 2019. 1

[17] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. Cell detection with star-convex polygons. *Medical Image Computing and Computer-Assisted Intervention*, pages 265–273, 2018. 2

[18] S.Q.Ren, K.M.He, R.Girshick, and J.Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 2015. 2

[19] W. Xu, H. Wang, F. Qi, and C. Lu. Explicit shape encoding for real-time instance segmentation. *ICCV*, pages 5168–5177, 2019. 2

[20] L. T. Y, M. M, and B. S. *et al*. Microsoft coco: Common objects in context. *ECCV*, 2014. 2, 4

[21] Z.Tian, C.Shen, H.Chen, and T.He. Fcos: Fully convolutional one-stage object detection. *ICCV*, 2019. 1, 3

[22] Z.Zhu, Q.Wang, B.Li, W.Wu, J.J.Yan, and W.M.Hu. Distractor-aware siamese networks for visual object tracking. *ECCV*, 2018. 1, 2