# FCOSIS: Fully-Convolutional One-Stage Instance Segmentation

Anonymous CVPR submission

Paper ID 8272

## Abstract

*We present FCOSIS, a fully convolutional single-stage anchor-free framework for instance segmentation. Our model predicts a class-agnostic, high-resolution segmentation map, similar to semantic segmentation, while preserving the benefits of multi-scale pyramidal networks for a high object detection performance. Instances are grouped in a per-pixel fashion using both the bounding box regression and a dense geometrical based embedding. Our design is conceptually simple, single-stage, and combines the benefits of object detection and semantic segmentation. On the MSCOCO dataset, we outperform competing single-shot approaches and are on par with more complex two-stage, anchor-based methods. Code will be made available upon acceptance.*

## 1. Introduction

Instance segmentation is the task that jointly estimates class labels and segmentation masks for all individual objects in an image. This is a fundamental goal in scene understanding and is an essential part in a variaty of different applications.

Historically, most current instance segmentation methods are extensions of prominent bounding box detectors. In recent years, the two most popular design principles for object detection are based on, so called, two-stage or one-stage approaches. Both design families rely on the usage of anchor boxes, which together with very deep backbone networks, have yielded state-of-the-art detectors while allowing for reasonably fast inference speeds. Anchor boxes discretize the input image space into a finite number locations with predefined locations, scales and aspect ratios. This discretization simplifies both training and testing, yet, has several disadvantages:

(i) parameters for size, aspect ratios and number of anchor boxes are usually hand-tuned ad-hoc heuristics, (ii) the detection of small objects with large shape variation is challenging, (iii) a large number of anchors is needed to yield high recall, (iv) the design is fundamentally different to methods for semantic segmentation, preventing principled approaches to solve for whole scene parsing.

Figure 1: FCOSIS works by dividing the task of instance detection and segmentation at different feature levels. A $(l, t, r, b)$ bounding box is predicted at every cell of a pyramidal feature network, while semantic segments are predicted in a high-resolution map and grouped with a geometrical based embedding, using a regression to the instance center (yellow arrow). (best viewed in color)

To circumvent these drawbacks, researchers have proposed object detection models that avoid the usage of anchor boxes. For the task of bounding box prediction, new anchor-free models, such as FCOS [21] and CenterNet [4], have demonstrated to perform on par with anchor-based methods, while being conceptually simpler and more intuitive. On small objects, they even show higher accuracy than anchor-based approaches.

While recent anchor free models are already competitieve in predicting bounding boxes, recent fully-convolutional per-pixel attempts for instance segmentation are fast but cannot compete with their anchor based counterparts in terms of accuracy [1].

In this paper, we want to bridge this gap and present an anchor-free single-shot model for instance mask prediction, which is conceptually simple but able to yield high quality mask predictions. Similar to popular anchor-based methods, such as, Mask R-CNN [8], our detector predicts both

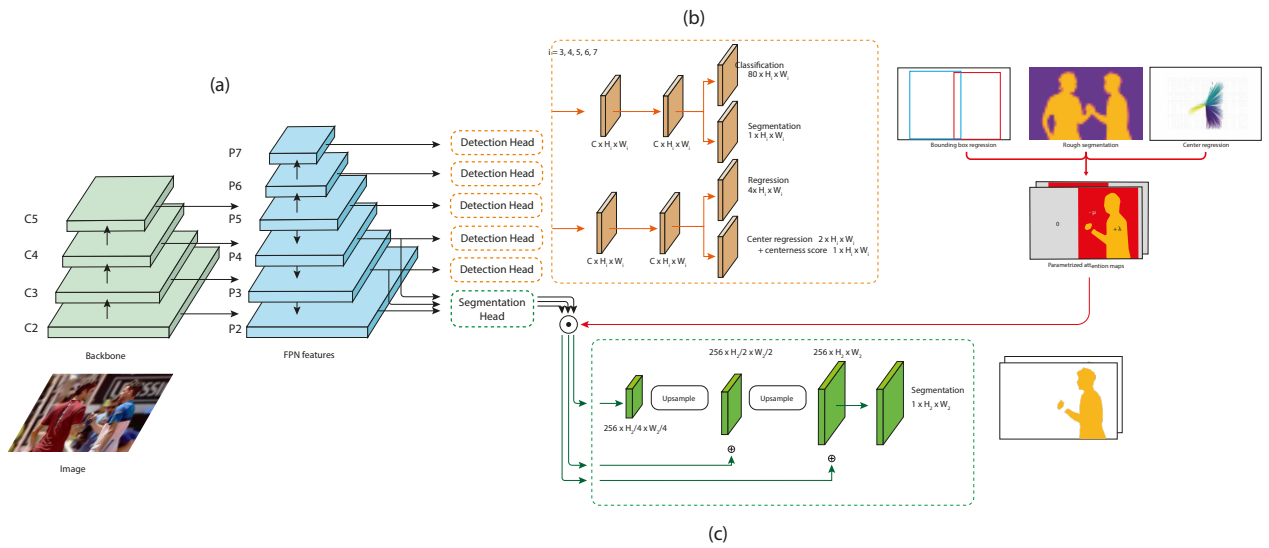


Figure 2: Architecture of FCOSIS. C2 to C5 denote the feature maps of the ResNet backbone network, P3 to P7 are the FPN feature maps.

bounding box locations, class predictions and instance segmentations of target objects, yet in an anchor-free, single-shot per-pixel manner. While our design extensions can be integrated to any fully-convolutional per-pixel based approach, in this paper, we showcase our method based on the popular FCOS object detector. Hence, we call our method Fully-Convolutional One-Shot Instance Segmentation (FCOSIS).

Besides having the benefit of richer object localization information through mask predictions, a second benefit of our method is that it allows to alleviate one challenging problem of anchor-free approaches, which is the correct sub-selection of pixel locations that are allowed to vote towards a specific target location. In FCOS, this task was approached by introducing a "centerness" parameter, where the distance to the bounding box center is regressed for every pixel. In our work, we show how to even increase this effect by leveraging the much richer information of segmentation masks. Using this both during training and inference leads to increased precision.

To this end, we introduce a fully-convolutional segmentation head operating at lower levels of the network. The separation of the segments into distinct target objects is supported by two additional prediction modules: The first module regresses centers of mass and provides a first signal about individual pixel instance memberships. This information is merged with a second module that provides low-resolutional whole-image foreground-background masks. Both cues are integrated into the fine-grained segmentation head via a top-down pathway.

Our contribution is four-fold: 1. We show that making use of mask information as a sole regularizer during training time already improves the performance of standard FCOS. 2. We demonstrate that adding mask prediction heads (but taking FCOS as a region proposal generator) in a two-stage apporach not only enables better segmentation mAP and inference speed compared to Mask R-CNN and RetinaMask [5], but also improves FCOS' bounding box mAP by a huge margin. 3. We design a novel one-stage instance segmentation architecture, based on two main strategies: First, we predict class, bounding box, mass centers and segmentation at every feature level, using a group sampling procedure. Second, we fuse these results via an attention map with features from lower levels to get sharp, high resolution instance segmentation maps, in a bottom-up manner. 4. We extensively test our novel design on publicly available datasets and show that we outperform single-stage, dense approaches and perform on-par with state-of-the-art two-stage instance segmentation networks, while being much simpler and more elegant.

## 2. Related Work

**Anchor-based object detection.** Most of the modern object detection methods are anchor-based, meaning that they rely on a dense grid of pre-defined bounding boxes needing to be classified (as positive or negative) and regressed (to refine their position and size). This design has been popularized by detectors like Faster R-CNN [7], SSD [16] or YOLOv2 [19] to name a few. While these methods perform at state-of-the-art level on well-studied datasets like MSCOCO [14], they need a careful tuning of several hyper-parameters, including anchor box shape, scale and distribution, as well as the way they are selected as

positive, negative or ignored during training. FreeAnchor [29] has proposed a method to discard this last point while remaining anchor-based, by a new loss function which assigns groundtruth detections to a certain anchor boxes. In MetaAnchor [26] the authors have show how to learn the all the anchor parameters during training.

**Anchor-free object detection.** YOLOv1 [18] is an example of anchor-free object detector. Instead of relying on anchor boxes, it directly regresses bounding boxes at points close to the object center. However, its design suffers from low recall since only the center pixel is used for bounding box prediction, making it perform poorly compared to its anchor-based follow up papers [19, 20].

Recently, several methods have appeared, discarding the necessity of anchors while performing at state-of-the-art for object detection. CornerNet [12] and CenterNet [4] regress a pair of corners and corners plus center, respectively. In a postprocessing step these anchor points are grouped using a feature embedding in a post-processing. FCOS [21] uses a much simpler design by densely predicting object bounding boxes for every pixel, in a semantic segmentation style.

Recently, anchor free approaches have also been shown beneficial for pedestrian detection [17] and even as an extension to anchor-based models [30].

**Instance segmentation.** Instance segmentation is the task to jointly estimate pixel-level class labels and masks for for every object instance in addition to the bounding box and class. As mentioned, most state-of-the-art approaches are extension of two-stage object detection approaches. For example, Mask R-CNN [8] and RetinaMask [5] first generate and crop object proposals using a ROI-Align module to repool feature maps and then separately estimate the mask, class and refined bounding-box in a second step. Several works ([5],) have shown that training an object detector to predict masks in a multi-tasking way not only enables to infer instance segmentations but also improves the bounding box precision.

SSAP [6] proposes a single-shot proposal-free instance segmentation method, based on the path-aggregation idea from [15]. This is done by predicting an affinity window for each grid cell which represents the probability that two pixels belong to the same instance. However, the post-processing step requires to solve a graph partitioning optimization problem which is NP-hard. In contrast, we demonstrate that a competitive one-stage instance segmentation method can be built without post-processing by leveraging the fully-convolutional nature of FCOS.

There have been late trends towards single-shot instance segmentation, with speed as their main motivation. For instance, recently YOLACT [1] was proposed, a method for instance segmentation that builds on the YOLO object detector [19]. The method, in parallel, generates a set of prototype masks and per-instance mask coefficients. Afterwards, the prototypes are linearly combined using the coefficients to yield mask predictions.

In [25], the authors show a method to infer points on the instance contour in a fully-convolutional fashion, which was extended by PolarMask [24] using the polar representation. Although these methods are fast, on MS COCO, they produce a 6-7 mAP accuracy drop in mask prediction compared to, e.g., Mask R-CNN.

In contrast, the recently proposed TensorMask [2], is able to produce higher accurate masks, yet still lower than Mask R-CNN. Additionally, as the method employs 4d tensor representations over multiple scales, this comes at a significantly increased complexity and memory cost. In this paper, we tend to strike a balance between the previously proposed methods. That is, we propose a method which is fast and conventually simple, yet delivers results that are on par or even higher than state-of-the-art two stage approaches!

**Panoptic segmentation.** Recently, a new field emerged which combines both instance and semantic segmentation, called Panoptic Segmentation [10]. Algorithms generate instance masks for all "foreground" objects and semantic segments for all "background" classes. Current state-of-the-art in this field is DeeperLab [27] and more recently Panoptic-DeepLab [3]. Although these methods deliver great results in panoptic quality most of them lack in sole instance segmentation, compared to state-of-the-art instance segmentation networks. In our work, we use ideas from panoptic segmentation in combination with anchor-free object detection heads to improve the mask prediction performance in a single stage.

## 3. FCOSIS

In this following section, we describe our approach of fully convolutional single-stage instance segmentation. First, we provide a short review of the FCOS object detection algorithm in Section 3.1. We then introduce in Section 3.2 the architecture of our approach for instance segmentation and discuss our the design choices in detail. Finally, in Section 3.3, we discuss how this architecture allows an improved per-pixel bounding box prediction using mask-regularization.

### 3.1. Fully Convolutional Object Detection

Although the ideas presented in our work are not bound to any specific fully-convolutional single-shot object detector, we decide to build on the recently proposed FCOS object detector [21], which we briefly review here.

Let $F_i \in \mathbb{R}^{H \times W \times F}$ be the the output at layer $i$ of the feature pyramid network (FPN). For a given training image, let $\{B^{(i)}\}$ be the set of groundtruth bounding-boxes, where $B_j^{(i)} = (x_0^{(i)}, y_0^{(i)}, x_1^{(i)}, y_1^{(i)})$ represent the left-top, right-

bottom coordinates of the $j^{\text{th}}$ groundtruth object and $c^{(i)} \in \{1, \ldots, C\}$ its corresponding class.

FCOS predicts bounding box locations in a per-pixel fashion. To this end, the network outputs 3 vectors at every location $(x, y)$ (and in every FPN layer $i$): the vector of classification scores $\boldsymbol{c}$ (of dimension $C$), the bounding box regression $\boldsymbol{t} = (l, t, r, b)$ and an additional "centerness" score $s$. The latter is used to down-weight pixel votes that are far from a potential target center.

If location $(x, y)$ falls into a groundtruth bounding box $B_j^{(i)}$, it is considered as a positive sample, in which case the targets are given by $c^*$, $\boldsymbol{t}^* = (l^*, t^*, r^*, b^*)$ and $s^* = \sqrt{\frac{\min(l^*,r^*)}{\max(l^*,r^*)} \times \frac{\min(t^*,b^*)}{\max(t^*,b^*)}}$; otherwise it is a negative sample and $c^* = 0$. Note that unlike traditional anchor-based object detection frameworks, the bounding box regressions do not designate offsets in position and scale but instead represent the distance to the left, top, right and bottom borders respectively, that is:

$$\begin{aligned} l^* &= x - x_0^{(i)}, \quad t^* = y - y_0^{(i)}, \\ r^* &= x_1^{(i)} - x, \quad b^* = y_1^{(i)} - y. \end{aligned} \tag{1}$$

FCOS is trained by backpropagating the following loss function:

$$\begin{aligned} L(\{\boldsymbol{p_{x,y}}\}, \{\boldsymbol{t}_{x,y}\}) = &\frac{1}{N_{\text{pos}}} \sum_{x,y} L_{\text{cls}}(\boldsymbol{c}_{x,y}, c^*_{x,y}) \\ &+ \frac{\alpha}{N_{\text{pos}}} \sum_{x,y} \mathbf{1}_{c^*_{x,y}>0} L_{\text{reg}}(\boldsymbol{t}_{x,y}, \boldsymbol{t}^*_{x,y}) \\ &+ \frac{\beta}{N_{\text{pos}}} \sum_{x,y} \mathbf{1}_{c^*_{x,y}>0} L_{\text{centerness}}(s, s^*), \end{aligned} \tag{2}$$

where the focal loss [22] is used as classification loss $L_{\text{cls}}$, the IoU loss $L_{\text{reg}}$ penalizes the bounding box regression, similar to UnitBox [28], and the binary cross entropy loss is used for $L_{\text{centerness}}$. $N_{\text{pos}}$ denotes the number of positive samples and the variables $\alpha$ and $\beta$ are used as trade-off parameters, which are set to 1 in all experiments.

## 3.2. One-Stage Instance Segmentation

In this section, we present the design of our Fully-Convolutional One-Shot Instance Segmentation network (FCOSIS) based on the pixel-wise object detection network FCOS. The full architectural overview is shown in Figure 2. In the following paragraph, we explain all the network components in detail.

### 3.2.1 Network Architecture

We start by noticing that object detection and pixel-wise segmentation have asymmetrical information level needs. Indeed, standard bounding box prediction heads of one-shot object detectors operate over pyramid levels, i.e., typically P3-P7, where larger objects are detected at higher levels and vice versa; as a consequence, objects detected at different levels have a similar relative size with respect to the resolution of their corresponding feature map. A natural extension towards segmentation prediction would be to classify pixels as foreground or background at the same level as for the detection task; however, doing this doesn't enable the creation of high-quality detailed segmentation masks from pixel-wise predictions, while using high resolution low-level feature maps would be a better choice. In turn, naively adding more low-level pyramid levels for the detection task has shown to lead to higher complexity and reduced accuracy for bounding box detection.

Hence, our design strategy proposes to exploit only the pyramid levels that have empirically shown to be a good choice for the object detection and pixel-level segmentation tasks, i.e., P3-P7 and P2-P4 respectively. The modules responsible for those two tasks are shown in Figure 2b and Figure 2c.

**Detection Head** We use the same model architecture at configuration as FCOS [21], i.e. four 3x3 `conv` layers of 256 channels with ReLU and GroupNorm, followed by a classification, regression and centerness prediction head.

**Segmentation Head** An overview of the segmentation head can be seen in Figure 2c. The module consumes the feature layers P2-P4 and yields a segmentation prediction having the same size as the P2 feature map, i.e., $W/4 \times H/4 \times C$, with $C$ being the number of object classes. The module uses three 3x3 `conv` layers with 256 channels and is followed by upsampling and summation operations.

In the context of instance segmentation though, the two modules described above need to be linked in order to localize and assign individual instance segments. We propose to do this by feeding the segmentation network with a parametrized attention map that can be seen as a instance prior or a top-down connection path for information fusion at lower network level. The FPN feature maps are then multiplied with the same attention map (bilinearly interpolated to match the corresponding resolution) and passed as input to the segmentation network. In the following, we discuss the construction of this attention map.

**Attention module** While the segmentation head above can output fine-grained segmentation masks, it lacks in global context to disambiguate neighboring objects and also needs further guidance to regions where higher accuracy is needed. To this end, we support the learning using a simple attention map, which is generated in upper layers and consists of three parameters: in particular, we set a zero value to pixels that are outside of the bounding box as they do not contain objects, and introduce two additional parameters that aim to help the network in separating objects close

to each other: $+\lambda$ and $-\mu$. $\lambda$ indicates foreground pixels on the object while $-\mu$ marks pixels which belong to the background or distractor instances. $\lambda$ and $\mu$ are initialized to 1 but are set as parameters that the network can learn.

In order to reliably predict these parameters in a fully-convolutional matter, we first generate coarse segmentations and a center of mass regression map. The segmentations are not good enough to produce high-quality segments but represent initial cues. The center regression module acts as an additional regularizer. All parameters are learned end-to-end during training.

The attention map generation clearer is constructed very simply from the bbox, coarse segm and center regression maps. For every segmentation forward pass (with GT or detected objects $\{1 \dots n\}$), each location $(x, y)$ gets assigned to a unique instance ID $1 \leq j \leq n$ whose center $(x_C^{(j)}, y_C^{(j)})$ is closest to the regressed center $(x + r_x, y + r_y)$. The grouped pixels are then multiplied with the coarse segmentation map and constrained to their bounding-box boundaries.

**Centerness of mass regression** In the pixel-wise object detection branch we can already regress if the pixel belongs to the foreground or the background. In order to help the network to group foreground pixels belonging to the same object we propose to use center regression. The goal of this module is to regress a vector pointing to the object center in addition to the FCOS centerness score. By grouping all vectors the belong to the same center we can separate pixels from neighboring objects even if they belong to the same class.

Figure 3: Center of mass regression map. Foreground pixels of masks are activated to regress the center of mass of the object they belong to. Colors are encoding the angle of the vector for a better visualization.

Formally, let $(x_C^{(j)}, y_C^{(j)})$ be the center of mass of the the $j^{\text{th}}$ groundtruth object. Then, at each location $(x, y)$, the center regression is defined as $(r_x, r_y) = (x_C^{(j)} - x, y_C^{(j)} - y)$. The center regression is learned using a SmoothL1 loss and activates only for positive samples, that is, pixels belonging to foreground objects.

**Efficient Attention Map Propagation** In order to increase both training and inference speed, we can leverage the fully convolutional nature of FCOSIS' segmentation branch in order to predict multiple instance segments in a single forward pass. A straight forward way to propagate the attention maps to the segmentation head is to create one map for each object; however, with a large number of detections, this becomes computationally expensive. In order to be more efficient in this step, we propose to combine several objects in one map by only requiring that the instances, to predict jointly, do not overlap. Formally, let $B^{(j)} = (x_0^{(j)}, y_0^{(j)}, x_1^{(j)}, y_1^{(j)})$ be the bounding box of the $j^{\text{th}}$ instance, $j \in \{1, \dots, N\}$ and let $o_{j,j'} = \text{IoU}(B^{(j)}, B^{(j')})$ designate the overlap between bounding boxes $j$ and $j'$. By noting $G = (V, E)$ the associated graph, where $V = \{1, \dots, N\}$ and $E = \{(i, j) \mid o_{i,j} > 0\}$, finding the minimum number of forward passes such that no bounding box overlaps reduces to determining the *chromatic number* $\chi(G)$ of $G$, which is defined as the smallest integer $q$ satisfying:

$$\begin{aligned} &\exists x = (x_v \in \{1, \dots, q\}, v \in V) \mid \\ &\quad \forall v, w \in V, (v, W) \in E \implies x_v \neq x_w, \end{aligned} \tag{3}$$

as shown in Figure 4

In practice, since the graph-coloring problem is NP-hard, we use the Largest-First (LF) heuristic, as described in [11] and [23].

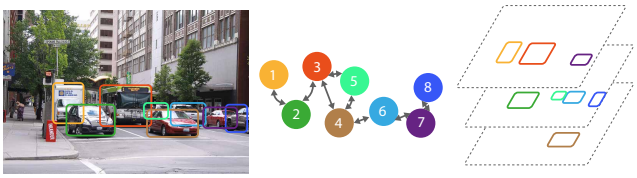


Figure 4: Graph coloring problem. Given a set of overlapping objects, find the minimal set of masks without bounding box overlaps.

### 3.2.2 Training

For a given input image, let $S \in \mathbb{R}^{H \times W}$ be the *class-agnostic* segmentation map prediction of the mask module and let $B^{(j)} = (x_0^{(j)}, y_0^{(j)}, x_1^{(j)}, y_1^{(j)})$ and $S^{(j)} \in \mathbb{R}^{H \times W}$ be respectively the groundtruth bounding box and binary segmentation map of the $j^{\text{th}}$ object. Since the ultimate goal of the predicted semantic segmentation map is only to obtain instance segments, we impose, for every object $j$, a Binary Cross Entropy loss at the cropped area corresponding to its groundtruth bounding box. Formally, this means that we

minimize:

$$L^{(j)}_{\text{segm}} = BCE(S_{y_0^{(j)}:y_1^{(j)},x_0^{(j)}:x_1^{(j)}}, S^{(j)}_{y_0^{(j)}:y_1^{(j)},x_0^{(j)}:x_1^{(j)}})$$

Since the groundtruth bounding box tightly delineates the object, we observe a class imbalance between foreground (1) and background (0) pixels using this method. To counter this issue, we add a padding of $p$ pixels on the left-right and top-bottom of the bounding box so that the ratio is approximately 0.5. Let $r = \frac{\sum \text{foreground pixels}}{w*h}$ be the ratio of foreground pixels in a bounding box of size $h \times w$. Then we want the new ratio to verify $r' = 0.5 = \frac{\sum \text{foreground pixels}}{(w+2p)\times(h+2p)}$. By solving this quadratic equation in $p$ we get the optimal padding, that we can round up to the nearest integer.

### 3.3. Mask regularization

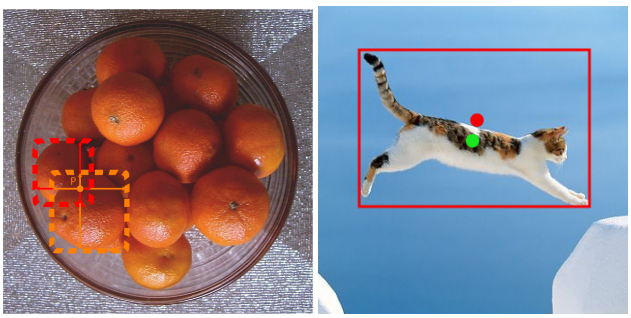

Figure 5: Left: Illustration of a highly ambiguous image. Since all objects have a similar scale, they are predicted in from the same FPN level. For instance, the groundtruth at point $P$ is assigned to the red bounding box since its area is smaller. while it would make more sense to attribute the groundtruth to the foreground object, e.g. the orange bounding box in this case. Right: Bounding box center (red) can be outside the object's real center. Center of mass (green) captures more information.

In [21], it is shown that FCOS' performance drops by almost 3 mAP if bounding box centerness prediction is not used as regularizer. In turn, it is also shown that the performance can be further significantly increased if the perfect ground-truth centerness is used. This insight suggests that the way positive samples are selected at training time and predictions are weighted during inference is suboptimal.

In particular, relying on bounding box centers can render ambiguity in cases where two target objects are close to each other (Fig 5,left) or during the detection of deformable objects (Fig 5,right), where the bounding box center is not guaranteed to be inside the object. For both cases, switching from bounding box center to more informative regularizations would be the preferable choice.

To this end, we propose – for ambiguous samples – to take advantage of the instance mask information at training time in order to favor objects in the foreground instead of objects of smallest area. Using masks additionally allows us to predict objects' center of mass, which is more accurate for deformable objects.

Specifically, if a location $(x, y)$ is *ambiguous* at feature level $i$, we choose the object whose segmentation label $s^{(i)}_{(x,y)}$ is the highest. Note that since a feature map $F_i$ (with stride $s$) has a downscaled resolution compared to the input image $I$, its segmentation label is not necessarily binary. Formally, if we denote $S^{(I)} \in \mathbb{R}^{H\times W} = \left(s^{(I)}_{i,j}\right)_{\substack{0\le i\le H-1\\ 0\le j\le W-1}}$ the segmentation map of a given object in the input image, the segmentation label $s^{(i)}_{x,y}$ at location $(x, y)$ in feature level $i$ is given by:

$$s^{(i)}_{x,y} = \frac{1}{s^2} \sum_{0\le i,j\le s-1} s^{(I)}_{xs+i,ys+j}$$

This segmentation information is beneficial in two ways:

1. The *ambiguity* problem is largely alleviated. Although, the COCO [14] dataset contains segmentation masks that overlap, each pixel is almost always mapped to a unique instance.

2. $L_{\text{reg}}$ can be better regularized. Instead of using the centerness scores as training weights, we can use a combination of centerness and segmentation. Specifically, we use:

   $$w = \alpha \cdot \text{centerness} + \beta \cdot s^{(i)}_{x,y} + \gamma \cdot \sqrt{\text{centerness} \times s^{(i)}_{x,y}}$$

   with $\alpha + \beta + \gamma = 1$. Intuitively, using the mask information enables to better handle cases of elongated objects whose foreground pixels are not necessarily concentrated in the center of their bounding box, as illustrated in figure 6.

(a) (b) (c)

Figure 6: Regularization maps using (a) $\alpha = 1,\ \beta, \gamma = 0$, (b) $\beta = 1,\ \alpha, \gamma = 0$ and (c) $\alpha, \beta, \gamma = 1/3$.

The above regularization approach can be applied if mask information is available during both training and inference, which is supported by FCOSIS.

## 4. Experiments

In an extensive evaluation we report the results on the large scale MSCOCO object detection and instance segmentation benchmark [14]. We use the 115k images from

| Method | Backbone | $AP^{bb}$ | $AP^{bb}_{50}$ | $AP^{bb}_{75}$ | $AP^{S}_{75}$ | $AP^{M}_{75}$ | $AP^{L}_{75}$ |
|---|---|---|---|---|---|---|---|
| Two stage detectors | | | | | | | |
| Mask R-CNN [8] | ResNet-50-FPN | 38.2 | 60.3 | 41.37 | 20.1 | 41.1 | 50.2 |
| RetinaMask [5] | ResNet-50-FPN | 39.4 | 58.6 | 42.3 | 21.9 | 42.0 | 51.0 |
| One-stage detectors | | | | | | | |
| YOLOv3-608 [20] | Darknet-53 | 33.0 | 57.9 | 34.4 | 18.3 | 35.4 | 41.9 |
| CornerNet [12] | Hourglass-52 | 37.8 | 53.7 | 40.1 | 17.0 | 39.0 | 50.5 |
| FCOS [21] | ResNet-50-FPN | 37.1 | 55.9 | 39.8 | 21.3 | 41.0 | 47.8 |
| FCOSreg (ours) | ResNet-50-FPN | 37.5 | 56.4 | 40.3 | 21.4 | 42.1 | 50.0 |
| MaskFCOS (ours) | ResNet-50-FPN | **39.8** | 57.9 | 43.2 | 23.2 | 43.8 | 52.5 |
| FCOSIS (ours) | ResNet-50-FPN | 38.2 | 56.2 | 42.1 | 22.7 | 42.3 | 51.2 |

Table 1: **Object detection Performance.** Comparision with state-of-the-art methods on COCO `test-dev`. FCOSreg deignates FCOS with mask regularization.

| Method | Backbone | AP | $AP_{50}$ | $AP_{75}$ | $AP^{S}_{75}$ | $AP^{M}_{75}$ | $AP^{L}_{75}$ |
|---|---|---|---|---|---|---|---|
| Two stage detectors | | | | | | | |
| Mask R-CNN [8] | ResNet-50-FPN | 34.9 | 57.2 | 36.9 | 15.4 | 36.6 | 50.8 |
| RetinaMask [5] | ResNet-50-FPN | 34.9 | 55.7 | 37.1 | 15.1 | 36.7 | 50.4 |
| One-stage detectors | | | | | | | |
| PolarMask [24] | ResNet-50-FPN | 29.1 | 49.5 | 29.7 | 12.6 | 31.8 | 42.3 |
| YOLACT [1] | ResNet-50-FPN | 28.2 | 46.6 | 29.2 | 9.2 | 29.3 | 44.8 |
| TensorMask [2]* | ResNet-50-3xFPN | 35.5 | 57.3 | 37.4 | 16.6 | 37.0 | 49.1 |
| MaskFCOS (ours) | ResNet-50-FPN | 34.8 | 55.3 | 37.4 | 15.7 | 37.9 | 51.9 |
| FCOSIS (ours) | ResNet-50-FPN | 32.4 | 51.5 | 34.3 | 12.9 | 35.2 | 49.2 |

Table 2: **Mask Performance.** Comparison with state-of-the-art methods on COCO `test-dev`. Methods marked with '*' use 3x more epochs to train.

the `trainval35k` split for training and the 5k images from the `minicval` split in our ablation studies. The final results are calculated on the larger `test-dev` split (20k images) by uploading them to the evaluation server. We report $AP$ as average precision of instance segmentation and $AP^{BB}$ for the bounding box precision.

**Training details.** FCOSIS is trained using the standard settings from the original FCOS paper. Images are resized to make the shorter side equal to 800 pixels while limiting the longer side to 1333 pixels. Unless specified, we use ResNet-50 [9] as backbone network in an FPN setting. The complete network is trained with Stochastic Gradient Descend (SGD) for 24 epochs with initial learning rate being 0.01 and a minibatch size of 16. The learning rate is reduced by a factor of 10 at epoch 14 and 21, with weight decay and momentum set to 0.0001 and 0.9, respectively.

**Mask Prediction Head** Here we show how one can extend FCOS [21] to predict instance masks using mask prediction heads (MaskFCOS). We select the top $N$ scoring bounding box predictions and use them as mask proposals. According to the equation 4 introduced in [13], we then sample features from a given feature map $P_k$ ($P_3$, $P_4$ or $P_5$), where $k$ is given by:

$$k = \left\lfloor k_0 + \log_2 \sqrt{wh}/224 \right\rfloor \tag{4}$$

Here, $k_0 = 4$ and $w, h$ are respectively the width and height of the bounding box prediction. In practice, this means that detections of size smaller than $224^2$ are assigned to $P_3$ and detections of size larger than $448^2$ are assigned to $P_5$, $P_4$ being used for the sizes in-between.

Similarly to [8], the ROI-Align operation outputs a 14x14 resolution feature map which is fed into four consecutive 3x3 convolutional layers and one 2x2 transposed convolutional layer. A final 1x1 convolutional layer then predicts the masks from the upsampled 28x28 feature map.

### 4.1. Comparison with state-of-the-art

We compare our model to state-of-the-art in the areas of bounding-box object detection and instance segmentation on MSCOCO `test-dev`, shown in Table 1 and Table 2, respectively. To make a fair comparison, we included both one-shot and two-shot approaches and tried to report

Figure 7: Sample instance segments of MaskFCOS (top) and FCOSIS (bottom) on MSCOCO images with *ResNet-50-FPN*. Notice how FCOSIS consistently produces sharper segments. The last three examples show failure cases where either a part of the instance is not segmented (the legs of the giraffe and the hand of the football player) or an object is not detected at all (the cat).

numbers with comparable backbone networks to *ResNet-50-FPN*.

### 4.2. Ablation study

In this section, we study the effects of different modules in our architecture and report quantitative results in Tab. 3.

**Top down connection** Following the segmentation branch architecture from DeeperLab [27], we build and evaluate a simple baseline without top-down connection: the segmentation output is of size $C \times H \times W$, where $C$ is the number of classes, and the segmentation branch contains an ASSP (Atrous Spatial Pyramid Pooling) module for an increased receptive field of the segmentation network. The segment outputs are then just cropped from this map using the predicted bounding box and class. As shown in table 3a, segmentation performance drops by over 3mAP, suggesting that distinguishing individual instances inside of bounding boxes (which is done via the instance-specific top-down connection) is crucial.

**Parametrized attention map** When setting the background / distractor pixels to zero in the parametrized attention (meaning that $\mu$ is set to 0 and is non-learnable), segmentation performance drops by 1.5 mAP (table 3b), which indicates that the negative features help to distinguish neighboring objects from one another.

**Segmentation & Center of mass regression** In the parametrized attention map, we evaluate the impact of the rough segmentation and center of mass regression. When using the bounding box information only, i.e. the entire

bounding box is filled with $\lambda$, segmentation performance is degraded slightly, meaning the high-level instance information captured by the detection branch can be leveraged in the lower-level segmentation branch.

| Method | AP | $AP_{50}$ | $AP_{75}$ | $AP^S_{75}$ | $AP^M_{75}$ | $AP^L_{75}$ |
|---|---|---|---|---|---|---|
| baseline | 28.9 | 48.9 | 29.6 | 10.6 | 30.8 | 43.5 |
| top-down | 32.4 | 51.5 | 34.3 | 12.9 | 35.2 | 49.2 |
| $\Delta$ | +3.5 | +2.6 | +4.7 | +2.3 | +4.4 | +5.7 |

(a) **Top-down connection**: cropping segments from a generic semantic segmentation map output vs using instance-specific information via the top-down connection of FCOSIS.

| Method | AP | $AP_{50}$ | $AP_{75}$ | $AP^S_{75}$ | $AP^M_{75}$ | $AP^L_{75}$ |
|---|---|---|---|---|---|---|
| $\lambda, \mu := 0$ | 30.9 | 50.8 | 32.0 | 12.6 | 33.1 | 46.2 |
| $\lambda, \mu$ | 32.4 | 51.5 | 34.3 | 12.9 | 35.2 | 49.2 |
| $\Delta$ | +1.5 | +0.7 | +2.3 | +0.3 | +2.1 | +3.0 |

(b) **Attention map parameters**: setting $\mu$ to 0 vs. setting it as a learnable parameter.

| Method | AP | $AP_{50}$ | $AP_{75}$ | $AP^S_{75}$ | $AP^M_{75}$ | $AP^L_{75}$ |
|---|---|---|---|---|---|---|
| bbox only | 31.5 | 50.9 | 32.9 | 12.7 | 34.5 | 47.4 |
| segm. | 32.4 | 51.5 | 34.3 | 12.9 | 35.2 | 49.2 |
| $\Delta$ | +0.9 | +0.6 | +1.4 | +0.2 | +0.7 | +1.8 |

(c) **Segmentation & Center of mass regression**: using only the bounding box information in the parametrized attention map vs taking additionally the rough segmentation and center regression into account.

Table 3: **Ablations** of FCOSIS on COCO `val2017` using *ResNet-50-FPN* and evaluating the segmentation performance.

### 4.3. Qualitative evaluation

We display the segmentation masks of our method FCOSIS and two other one-stage instance segmentation methods in Figure 8. Since YOLACT [1] and FCOSIS are based on a segmentation map output, we observe sharp segments for both methods, in comparison to PolarMask [24]. However, FCOSIS has a stronger object detection performance, resulting in instances being better detected.

At the time of writing, the implementation of TensorMask [2] is not available and therefore cannot be compared here.

## 5. Conclusion

In this paper, we have introduced FCOSIS, a novel method for fully convolutional one-stage instance segmentation. In contrast to previous methods, we keep the simplicity of current one-shot approaches while being competitive to more complex state-of-the-art two-stage anchor based methods.

To allow for both dense per-pixel high-resolution segmentation maps and accurate bounding box predictions, FCOSIS links lower and higher level cues using a parameterized attention map. The map, in turn, is created by fusing coarse segmentations, bounding box regressions and center of mass predictions.

In contrast to anchor based approaches, for FCOSIS less parameters need to be optimized and the design is closer to typical semantic segmentation approaches. This renders an obvious extension of our method to the task of panoptic segmentation, which we keep as future work.

## References


[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: real-time instance segmentation. In *Proc. ICCV*, 2019. 1, 3, 7, 9, 10

[2] Xinlei Chen, Ross B. Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *Proc. ICCV*, 2019. 3, 7, 9

[3] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab, 2019. 3

[4] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proc. ICCV*, October 2019. 1, 3

[5] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C Berg. Retinamask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv preprint arXiv:1901.03353*, 2019. 2, 3, 7

[6] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. *Proc. ICCV*, 2019. 3

[7] Ross Girshick. Fast r-cnn. In *Proc. ICCV*, 2015. 2

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. ICCV*, 2017. 1, 3, 7

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 7

[10] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *Proc. CVPR*, June 2019. 3

[11] Adrian Kosowski and Krzysztof Manuszewski. Classical coloring of graphs. In *Contemporary Mathematics*, 2004. 5

[12] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proc. ECCV*, 2018. 3, 7

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017. 7

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014. 2, 6

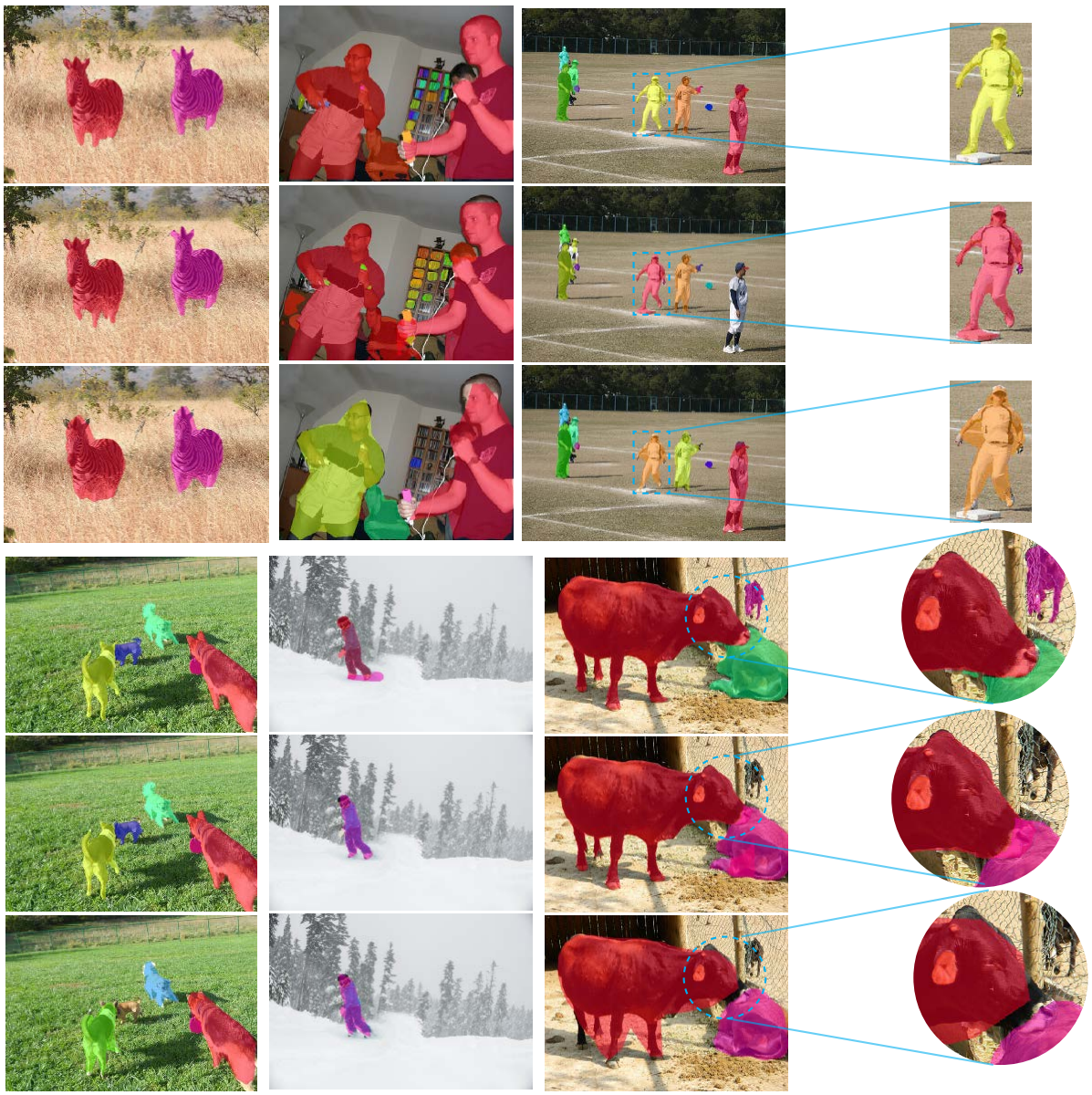

Figure 8: Visual comparison of our method FCOSIS (top) and two other one-stage instance segmentation methods on COCO `val2017`: YOLACT [1] (middle) and PolarMask [24] (bottom) with *ResNet-50-FPN*. Notice how PolarMask produces very rough segmentations compared to the two other methods and how FCOSIS consistently detects instances better than YOLACT.

[15] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proc. ECCV*, 2016. 2

[17] Wei Liu, Shengcai Liao, Weiqiang Ren, Weidong Hu, and Yinan Yu. High-level semantic feature detection: A new perspective for pedestrian detection. In *Proc. CVPR*, pages 5187–5196, 2019. 3

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016. 3

[19] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proc. CVPR*, July 2017. 2, 3

[20] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, 2018. 3, 7

[21] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos:

Fully convolutional one-stage object detection. *Proc. ICCV*, 2019. 1, 3, 4, 6, 7

[22] Lin Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. ICCV*, 2017. 4

[23] D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 1967. 5

[24] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. *arXiv preprint arXiv:1909.13226*, 2019. 3, 7, 9, 10

[25] Wenqiang Xu, Haiyang Wang, Fubo Qi, and Cewu Lu. Explicit shape encoding for real-time instance segmentation. In *Proc. ICCV*, October 2019. 3

[26] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In *Proc. NeurIPS*, 2018. 3

[27] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019. 3, 8

[28] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object deteciton framework. In *Proceedings of the International Conference on Multimedia ACM*, October 2016. 4

[29] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. *Proc. NeurIPS*, 2019. 3

[30] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. *Proc. CVPR*, 2019. 3