# Deeply Shape-guided Cascade for Instance Segmentation

Hao Ding[1], Siyuan Qiao[1], Alan Yuille[1], Wei Shen[2](✉)
[1]Department of Computer Science, Johns Hopkins University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
{hding15,siyuan.qiao,ayuille1}@jhu.edu, wei.shen@sjtu.edu.cn

## Abstract

*The key to a successful cascade architecture for precise instance segmentation is to fully leverage the relationship between bounding box detection and mask segmentation across multiple stages. Although modern instance segmentation cascades achieve leading performance, they mainly make use of a unidirectional relationship, i.e., mask segmentation can benefit from iteratively refined bounding box detection. In this paper, we investigate an alternative direction, i.e., how to take the advantage of precise mask segmentation for bounding box detection in a cascade architecture. We propose a Deeply Shape-guided Cascade (DSC) for instance segmentation, which iteratively imposes the shape guidances extracted from mask prediction at previous stage on bounding box detection at current stage. It forms a bi-directional relationship between the two tasks by introducing three key components: (1) Initial shape guidance: A mask-supervised Region Proposal Network (mPRN) with the ability to generate class-agnostic masks; (2) Explicit shape guidance: A mask-guided region-of-interest (RoI) feature extractor, which employs mask segmentation at previous stage to focus feature extraction at current stage within a region aligned well with the shape of the instance-of-interest rather than a rectangular RoI; (3) Implicit shape guidance: A feature fusion operation which feeds intermediate mask features at previous stage to the bounding box head at current stage. Experimental results show that DSC outperforms the state-of-the-art instance segmentation cascade, Hybrid Task Cascade (HTC), by a large margin and achieves 51.8 box AP and 45.5 mask AP on COCO* `test-dev`*. The code is released at:* https://github.com/hding2455/DSC.

## 1. Introduction

Instance segmentation [1, 43, 44, 21, 14], an increasingly active research topic in recent years, is a combination of the elements from two classical computer vision tasks - object detection [19, 20, 46, 35, 45, 17, 31, 60] and semantic segmentation [37, 12, 61, 59, 38, 22]. It is challenging since it requires not only classifying and localizing all the object instances correctly in an image, but also providing a precise segmentation mask for each instance at the same time.


Figure 1: Instance segmentation on huddled instances.

To achieve precise instance segmentation, building a cascaded architecture [5] with multi-stage refinement is a promising strategy. As pointed out in [9], the key to a successful cascade architecture for precise instance segmentation is to fully leverage the relationship between bounding box detection and mask segmentation across multiple stages. Surprisingly, we find that leading instance segmentation cascades [6, 9], although achieving state-of-the-art performances, mainly make use of a unidirectional relationship, *i.e.*, mask segmentation can benefit from iteratively refined bounding box detection. In this paper, we investigate the opposite direction, *i.e.*, how to take the advantage of precise mask segmentation for bounding box detection. Our aim is to establish a bi-directional relationship between bounding box detection and mask segmentation in a cascade architecture to boost the instance segmentation performance.

Towards this end, we propose a **D**eeply **S**hape-guided **C**ascade (DSC) for instance segmentation, which iteratively imposes the shape guidances extracted from mask prediction at previous stage on bounding box detection at current stage. DSC plugs the intuitions of Deeply Supervised Nets (DSN) [26, 54], *i.e.*, 1) enforcing early supervision for intermediate stages and 2) fusing features across multiple stages benefits representation learning, into an instance segmentation cascade. This leads to three key components for shape guidance learning: (1) Initial shape guidance: A mask-supervised Region Proposal Network (mPRN) with

✉ Corresponding Author.

the ability to generate class-agnostic masks; (2) Explicit shape guidance: A mask-guided region-of-interest (RoI) feature extractor, based upon the segmented mask at previous stage, focuses on extracting features within a region aligned with the shape of the instance-of-interest rather than a rectangular RoI at current stage; (3) Implicit shape guidance: A feature fusion operation which feeds intermediate mask features at previous stage to the bounding box head at current stage. The shape guidances, either explicitly or implicitly learned from mask-level supervision, are more informative than box-level supervision, and thus are able to help generate more precise bounding boxes. Then, these bounding boxes can lead to more accurate segmented masks. This forms a positive feedback loop between bounding box detection and mask segmentation across multiple stages in the cascade, facilitating achieving precise instance segmentation.

DSC is easy to implement and can be trained end-to end. Without bells and whistles, on average, it consistently outperforms the state-of-the-art instance segmentation cascade, HTC [9], with different backbones by 2.1 box AP and 1.5 mask AP on COCO 2017 `val` and 1.9 box AP and 1.4 mask AP on COCO 2017 `test-dev`, thanks to the positive feedback loop between mask prediction and bounding box detection. It is worth mentioning that DSC is good at segmenting huddled instances, *i.e.*, instances that are crowded together as shown in Fig.1, benefited from the shape guidances. We carefully select a subset of COCO 2017 `val` which contains a high portion of huddle instances. DSC achieves significant improvements compared with other methods on this subset.

To sum up, our main contribution is the proposal of a new cascade architecture for precise instance segmentation. It explores a different direction to leverage the relationship between bounding box detection and mask segmentation, forming a positive feedback loop between the two tasks by introducing shape guidances into bounding box detection. It achieves consistent and substantial improvements over the state-of-the-art instance segmentation cascade, Hybrid Task Cascade (HTC) [9], on the COCO dataset.

## 2. Related Work

### 2.1. Non-cascade Instance Segmentation

Since instance segmentation combines object detection and semantic segmentation, existing methods for this task can be roughly categorized into two types: segmentation-based and detection-based.

#### 2.1.1 Segmentation-based Methods

Segmentation-based methods usually adopt a two-step paradigm - "segment then identify", *i.e.*, first perform semantic segmentation to obtain a per-pixel category-level segmentation map for an image, and then identify each object instance therefrom. Liang *et al.* [30] proposed to identify object instances from the segmentation map by spectral clustering. Kirillov *et al.* [25] partitioned instances from the segmentation map with the help of a learned instance-ware edge map under a MultiCut formulation. Arnab and Torr [2] made use of the cues from the output of an object detector to identify instances from the segmentation map. Zhang *et al.* [58] predicted instance labels for local patches and merged similar predictions via a Markov Random Field (MRF). They then improved this method by using a densely connected MRF instead, which exploits fast inference [57]. Wu *et al.* [52] proposed a Hough-like Transform to bridge category-level and instance level segmentation, while Bai and Urtasun [3] achieved this by Watershed Transform. There are also some other methods that form instances from a segmentation map by learning an embedding to group similar pixels [40, 22, 39]. Liu *et al.* [33] broke the grouping problem into a series of sub-grouping problems and addressed sequentially.

#### 2.1.2 Detection-based Methods

Detection-based methods first generate candidate bounding boxes, then segment the instance mask from each of them. Depending on how to generate the candidate bounding boxes, detection-based methods can be categorized into two classes: anchor-free and anchor-based.

**Anchor-free methods.** The early work of anchor-free methods directly used dense sliding-windows as the candidate bounding boxes, such as DeepMask [43], SharpMask [44] and InstanceFCN [16], which applied convolutional neural networks to predicting object masks in a dense sliding-window manner. Recent anchor-free methods design more sophisticated to generate mask proposals. YOLACT [4] first learned a dictionary of mask prototypes and then predicted per-instance coefficients to linearly combine prototypes to produce an instance mask. ExtremeNet [62] used keypoint detection to predict extreme points, which provide an octagonal approximation for an instance mask. PolarMask [53] built a polar representation for each instance mask and formulated instance segmentation as instance center classification and dense distance regression in a polar coordinate. TensorMask [13] revisited the paradigm of dense sliding window instance segmentation and represented masks by structured 4D tensors over a spatial domain. SOLO [48] and its upgraded version [51] covert instance segmentation into a classification problem by assigning categories to all pixels inside an instance according to its position and scale. Deep Snake [42] and PolyTransform [29] predicted position offsets w.r.t. the vertices of the polygonal contour of the mask. CondInst [47] estimated the mask head by conditional convolution kernels [55] to make discriminative mask predictions and eliminate feature alignment. These anchor-free methods mainly focus on real-time performance, and the high precision of the results is not their first priority.

**Anchor-based methods.** Anchor-based methods take anchors as references to predict region proposals as the candidate bounding boxes, and then segment each instance mask using the box as a guide [28, 11]. This paradigm is known as "detect then segment", which is currently the dominant paradigm. Mask-RCNN is a representative instantiation of this paradigm, which extended the well-known anchor-based object detector, Faster R-CNN [46], with a mask segmentation branch. Follow-up works, *i.e.*, the variants of Mask-RCNN, improved it by enhancing feature pyramid with accurate localization signals existed in low-level layers [34], re-scoring the confidence of a predicted mask to calibrate the misalignment between the mask score and its localization accuracy [24, 15], or using a more sophisticated bounding box regression method [7].

### 2.2. Cascade Instance Segmentation

Cascade architectures emerge recently along with the increasing demand for precise object detection. CRAFT [56] built a cascade structure for both Region Proposal Network [46] and Fast R-CNN [20] to get higher quality proposals and detection results. CC-Net [41] excluded easy negative samples at early stages in a cascade. Li *et al.* [27] introduced a CNN cascade that operates at multiple resolutions for face detection.

As far as we know, there are only two cascade architectures for instance segmentation, *i.e.*, Cascade Mask R-CNN [6] and Hybrid Task Cascade (HTC) [9]. As pointed out in [9], it is nontrivial to integrate the idea of cascade into instance segmentation. For example, a simple combination of Cascade R-CNN [5] and Mask R-CNN [23], *i.e.*, Cascade Mask R-CNN [6], which iteratively feeds the refined bounding boxes at current stage into next one as high-quality RoIs, only leads to limited gain. HTC [9] improves Cascade Mask R-CNN by connecting the mask heads at multiple stages through mask information flow.

Both HTC and Cascade Mask R-CNN deliver the message that the mask prediction branch can benefit from the updated bounding box regression. Our method, DSC, shows an orthogonal direction: the object detection branch, *i.e.*, object classification and bounding box regression, can also take advantage of the mask predictions, leading to a positive feedback loop between bounding box detection and mask segmentation.

## 3. Methodology

In this section, we first introduce the overall framework of Deeply Shape-guided Cascade (DSC) for instance segmentation, then elaborate on the three newly introduced key components: the mask-supervised Region Proposal Network (mRPN), the mask-guided ROI feature extractor (shape-guided RoIAlign), and the feature fusion operation. They impose initial, explicit, and implicit shape guidances, respectively, on bounding box detection in the cascade.

### 3.1. Overall Framework

The overall framework of DSC is shown in Fig. 2. It follows the cascade paradigm [6, 9], *i.e.*, first generating a set of instance proposals by a Region Proposal Network (RPN), then iteratively refining the bounding boxes of the proposals and segmenting masks from them by a sequence of box and mask heads. DSC has three new key components:

- We replace the RPN with the mask-supervised RPN (mRPN), which is guided by both the box supervision $\mathcal{B}_g$ and the mask supervision $\mathcal{M}_g$. Given the feature map $\mathbf{F}$ produced by a CNN backbone as the input, the mRPN produces not only a set of RoIs $\mathcal{B}^0$ but also class-agnostic mask probability maps $\mathcal{M}^0$ corresponding to these RoIs. In addition, it also outputs a set of intermediate mask feature maps $\mathcal{F}^0_{\_}$. Let $\mathbf{B}^0 \in \mathcal{B}^0$ be an RoI, then $\mathbf{M}^0 \in \mathcal{M}^0$ and $\mathbf{F}^0_{\_} \in \mathcal{F}^0_{\_}$ are its corresponding mask probability map and intermediate mask feature map, respectively. This component involves **initial shape guidance**, as the shape guidance is learned from the mask supervision and imposed on the early stage (proposal generation stage) of the cascade.
- For all the box heads in the cascade, we replace the feature extractor, *i.e.*, RoIAlign, with the mask-guided ROI feature extractor, *i.e.*, shape-guided RoIAlign. Shape-guided RoIAlign employs the mask predictions $\mathcal{M}^{t-1}$ at stage $t-1$ to focus feature extraction at stage $t$ within a region aligned well with the shape of the instance-of-interest. This component involves **explicit shape guidance**, as the learned shape guidance (mask prediction) is directly applied to feature extraction.
- We conduct a feature fusion operation to enhance the input features of the box head at stage $t$ by integrating them with the intermediate mask feature maps $\mathcal{F}^{t-1}_{\_}$ at stage $t-1$. This component involves **implicit shape guidance**, as the intermediate mask features are indirectly learned from the mask supervision.

The details of the three components will be described in the following sub-sections.

### 3.2. Mask-supervised RPN

The detailed design of the mask-supervised RPN (mPRN) is shown in Fig. 3. The mRPN consists of two parts: The first part is the same as the RPN and the second part is a class-agnostic mask generator supervised by the mask supervision $\mathcal{M}_g$, named Mask Proposal Network (MPN). Let $\mathcal{B}^0$ be RoIs outputted by the first part, their feature maps $\mathcal{F}^0_m$ are computed by a RoIAlign layer, which serves as the input of the MPN to generate class-agnostic mask probability maps $\mathcal{M}^0$ within $\mathcal{B}^0$. Both $\mathcal{B}^0$ and $\mathcal{M}^0$ are the outputs of the mPRN, and it also preserves intermediate feature maps $\mathcal{F}^0_{\_}$ of the MPN for further feature fusion.

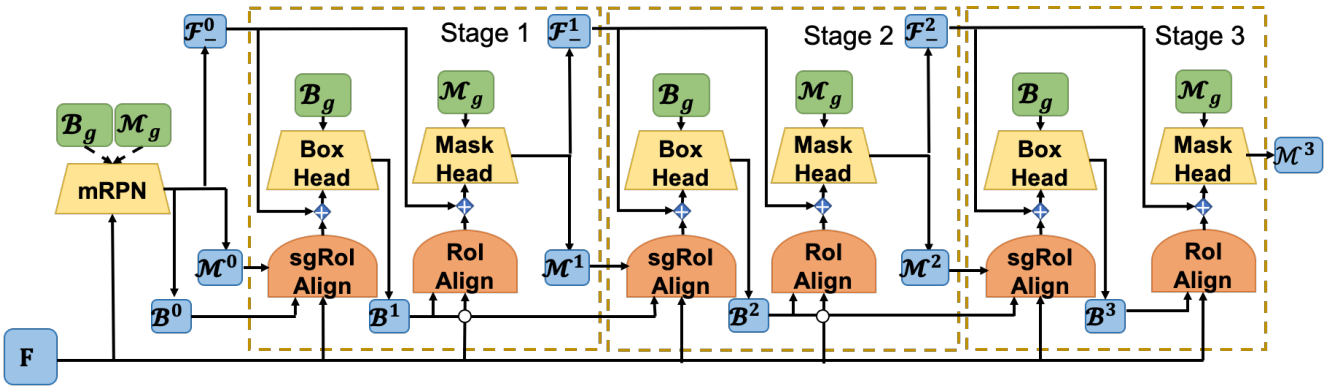


Figure 2: The overall framework of Deeply Shape-guided Cascade (DSC) for instance segmentation. Please refer to the first paragraph of Sec. 3.1 for the meanings of the notations.

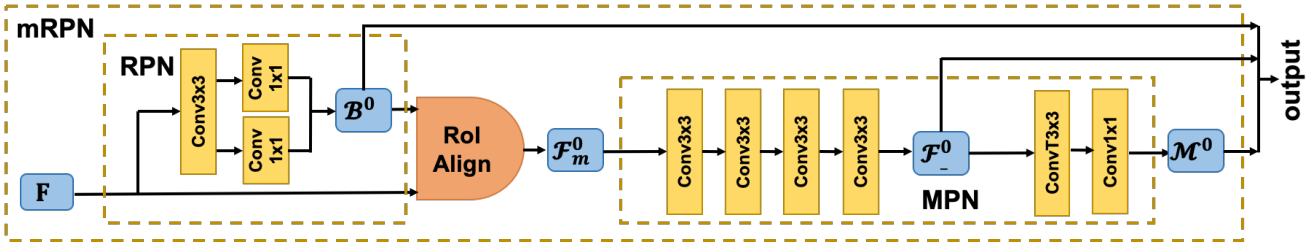


Figure 3: The detail of mask-supervised RPN (mRPN). The supervision flow is omitted for illustration simplicity.

### 3.3. Shape-guided RoIAlign

Shape-guided RoIAlign (SgRoIAlign) computes feature values under the guidance of the probability of being the instance-of-interest at each location, *e.g.*, a predicted mask probability map $\mathbf{M}$. Similar to RoIAlign, it first divides a RoI $\mathbf{B}$ into $H \times W$ bins. Each bin is denoted by $R_{h,w} = \{(x_h^1, y_w^1), (x_h^2, y_w^2)\}$, where $(x_h^1, y_w^1)$, $(x_h^2, y_w^2)$ are the continuous coordinates of the top-left and bottom-right points of the bin at the $h^{th}$ row and $w^{th}$ column, respectively. $N$ sampling points at continuous location $\{(a_{w,h}^i, b_{w,h}^i)\}_{i=1}^N$ are located uniformly within this bin $R_{w,h}$ for feature extraction. Then, given the $H_p \times W_p$ mask probability map $\mathbf{M}$ aligned with the RoI $\mathbf{B}$, where each element $m(j,k)$ denotes the probability of being the instance-of-interest at a discrete location $(j,k)$, and a sampling point at location $(a_{h,w}^i, b_{h,w}^i)$ on the feature map $\mathbf{F}$, we can compute the corresponding location $(c_{h,w}^i, d_{h,w}^i)$ at probability map $\mathbf{M}$ by equations:

$$c_{h,w}^i = (a_{h,w}^i - x_1^1) \times \frac{H_p}{H},\ d_{h,w}^i = (b_{h,w}^i - y_1^1) \times \frac{W_p}{W} \tag{1}$$

The feature value $f(a_{h,w}^i, b_{h,w}^i)$ at location $(a_{h,w}^i, b_{h,w}^i)$ on the feature map $\mathbf{F}$ and the probability value $m(c_{h,w}^i, d_{h,w}^i)$ at the corresponding location $(c_{h,w}^i, d_{h,w}^i)$ on the mask probability map $\mathbf{M}$ are computed by bi-linear interpolation.

Then, the feature representation $f_{\mathbf{B},\mathbf{M}}(h,w)$ of a bin $R_{h,w}$ is obtained by averaging the multiplications between feature values and probability values plus one at the same sampling points:

$$f_{\mathbf{B},\mathbf{M}}(h,w) = \sum_{i=1}^{N} \frac{f(a_{h,w}^i, b_{h,w}^i) \times (1 + m(c_{h,w}^i, d_{h,w}^i))}{N}. \tag{2}$$

The intuition of Eq. 2 is two-fold. On one hand, we want to decrease the impact of the context features and focus feature extraction within the predicted shape region. On the other hand, we do not want to totally exclude the context features, since they are also helpful for object recognition [36] and we cannot guarantee that the predicted shape region is perfect. A diagram to illustrate shape-guided RoIAlign is provided in the supplementary material.

Finally, we obtain the small feature map $\mathbf{F}_{\mathbf{B},\mathbf{M}}$ by repeating the above computation for each bin. We denote this feature extraction procedure, shape-guided RoIAlign, by a function $\mathbf{F}_{\mathbf{B},\mathbf{M}} = \mathbb{f}_s(\mathbf{B}, \mathbf{M}, \mathbf{F})$. Correspondingly, the vanilla RoIAlign procedure can be denoted as function $\mathbf{F}_{\mathbf{B}} = \mathbb{f}(\mathbf{B}, \mathbf{F})$. Note that in our cascade, as shown in Fig. 2, shape-guided RoIAlign and RoIAlign are used to compute the features for box heads and mask heads, respectively. Thus, we rewrite $\mathbf{F}_b = \mathbf{F}_{\mathbf{B},\mathbf{M}}$ and $\mathbf{F}_m = \mathbf{F}_{\mathbf{B}}$ for notational clearness.

### 3.4. Feature Fusion Operation

The intermediate mask features implicitly learned from the more informative mask supervision at current stage can

guide box prediction and mask prediction at next stage. We feed an intermediate feature map $\mathbf{F}^t_-$ into a $1 \times 1$ convolutional layer, then fuse it with the box features $\mathbf{F}^{t+1}_b$ or the mask features $\mathbf{F}^{t+1}_m$ by the element-wise summation. This fusing operation is not trivial since the predicted boxes (RoIs) are iteratively refined in the cascade, which results in feature misalignment among stages.

To eliminate this misalignment issue, recomputing the mask features for the updated RoIs of each stage is a solution, but computationally expensive. The total number of convolutional operations to recompute the mask features is a quadratic function of the total number of the stages.

We propose a simple yet effective strategy for adaptive feature alignment. The basic idea is to reuse the intermediate features of previous stages based on two empirical facts. (1) Extracting features on the enlarged regions of RoIs with a fixed and relatively small enlargement ratio ($\leq 2$) does not degrade the overall performance of the cascade. An enlarged region of an RoI is a rectangular region that shares the same center of the RoI, but the width and height are enlarged by a scale factor $r$. (2) After bounding box regression, most of RoIs are still in the enlarged regions, and clipping those out-of-region RoIs into the enlarged regions does not degrade the performance of the cascade.

We provide two studies on the cascade Mask R-CNN model to demonstrate these two facts. For the first one we enlarge RoIs for feature extraction with decreasing enlargement ratios $2^{\frac{3}{3}}, 2^{\frac{2}{3}}, 2^{\frac{1}{3}}$ for stage $1, 2, 3$ of the cascade Mask R-CNN model respectively. This modified cascade Mask R-CNN model achieves competitive performance regarding both box AP (41.3 vs 41.2) and mask AP (36.0 vs 35.9) comparing to the original version. For the second one we choose different enlargement ratios $2^{\frac{3}{3}}, 2^{\frac{2}{3}}, 2^{\frac{1}{3}}$ to generate enlarged regions for RoIs and calculate the percentage of stay-in-region RoIs, after bounding box regression. We find 98.3%, 89.16%, 64.9% of the RoIs are in the enlarged region after regression. We also clip all out-of-region RoIs and ensure that they are within their corresponding enlarged regions, and observe no AP drop by doing this.

The adaptive feature alignment strategy is designed based on these two empirical facts. It consists of two vital steps: 1) RoI enlargement with decreasing enlargement ratios for feature extraction from early to late stages in the cascade; 2) RoI clipping to ensure the enlarged region of each RoI for the following stage is still within the enlarged region of this stage after bounding box regression. Then the features extracted from the enlarged RoI can be reused at next stage without recomputing. Fig. 4 illustrates this strategy. $\mathbf{B}^t$ is an RoI at stage $t$ in the cascade and $\mathbf{B}^t_e$ is its enlarged region. Feature extraction at stage $t$ is performed on $\mathbf{B}^t_e$ and the intermediate mask features are retained. At stage $t+1$, $\mathbf{B}^t$ is regressed to $\mathbf{B}^{t+1}$, whose enlarged region $\mathbf{B}^{t+1}_e$ is guaranteed to be within $\mathbf{B}^t_e$ by clipping. Then the mask features for stage $t+1$ can be directly obtained from the retained intermediate mask features at stage $t$. We denote this adaptive feature alignment procedure as a function: $\mathbf{F}^{t+1}_{a-} = \mathbb{a}(\mathbf{F}^t_-, \mathbf{B}^t_e, \mathbf{B}^{t+1}_e)$, where $\mathbf{F}^t_-$ is the intermediate mask feature map retained at stage $t$ which is not aligned with RoI $\mathbf{B}^t$ and the output $\mathbf{F}^{t+1}_{a-}$ is the aligned intermediate mask feature map.

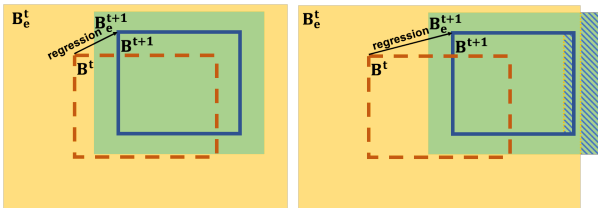


Figure 4: Illustration of the adaptive feature alignment strategy. At stage $t$, feature extraction is performed on the yellow region $\mathbf{B}^t_e$, which is an enlarged region of the original RoI $\mathbf{B}^t$. At stage $t+1$, $\mathbf{B}^t$ is regressed to $\mathbf{B}^{t+1}$. If the enlarged region of $\mathbf{B}^{t+1}$, *i.e.*, the green region $\mathbf{B}^{t+1}_e$, is still within $\mathbf{B}^t_e$, we keep $\mathbf{B}^{t+1}$, as shown in the left figure; Otherwise, we clip a part of $\mathbf{B}^{t+1}$ (the shadow area) to ensure its enlarged region is within $\mathbf{B}^t_e$, as shown in the right figure.

This strategy allows us to extract aligned intermediate features from the retained intermediate features of the previous stage without recomputing. Therefore, the number of convolutional operations for computing the mask features remains a linear function of the total number of the stages.

### 3.5. Cascade Pipeline

Now we give the formula to summarize the cascade pipeline. At stage $t$, let $\mathbb{b}^t$ and $\mathbb{m}^t$ denote the functions of the box head and the mask head, respectively, then we can write the DSC pipeline as:

$$\begin{aligned} \mathbf{F}^t_m &= \mathbb{f}(\mathbf{B}^t_e, \mathbf{F}), \\ (\mathbf{M}^t, \mathbf{F}^t_-) &= \mathbb{m}^t(\mathbf{F}^t_m \oplus \mathbf{w}^t_m \mathbb{a}(\mathbf{F}^{t-1}_-, \mathbf{B}^{t-1}_e, \mathbf{B}^t_e)), \\ \mathbf{F}^{t+1}_b &= \mathbb{f}_s(\mathbf{B}^t_e, \mathbf{F}, \mathbf{M}^t), \\ \mathbf{B}^{t+1} &= \mathbb{b}^t(\mathbf{B}^t, \mathbf{F}^{t+1}_b \oplus \mathbf{w}^{t+1}_b \mathbf{F}^t_-), \end{aligned} \tag{3}$$

where $\oplus$ denotes the element-wise summation operator, and $\mathbf{w}^t_m$ and $\mathbf{w}^t_b$ are the weights of the $1 \times 1$ convolutional layers for the box head and mask head at stage $t$, respectively, to process the intermediate mask feature map $\mathbf{F}^{t-1}_-$. Note that, stage 0 of the DSC is an mRPN, *i.e.*, $\mathbf{B}^0$, $\mathbf{M}^0$ and $\mathbf{F}^0_-$ are produced by the mRPN. Note that, the default resolution of the intermediate mask feature maps is $14 \times 14$. We can reduce the resolution to $7 \times 7$ to speed up the cascade. This fast version of DSC is denoted by F-DSC.

## 4. Experimental Result

All experiments are conducted on the COCO 2017 dataset [32], which contains about 118k images with corresponding annotations as the training set and 5k held-out images with annotations as the validation set. The main

metric used for evaluation is the standard COCO-style Average Precision (AP) averaged across IoU thresholds from 0.5 to 0.95 with 0.05 as the interval. Both box AP ($AP_b$) and mask AP ($AP_m$) are evaluated. We also report $AP^{50}$ and $AP^{75}$ (AP at IoU threshold $= 0.5$ and $= 0.75$). Our models are trained on the 118k training set. Results on the held-out 5k validation set and the 20k test-dev set are reported.

### 4.1. Implementation Details

We use *mmdetection* [10] as the codebase. For fair comparison. The baselines, *e.g.*, Cascade Mask R-CNN and HTC, are also implemented by the same codebase.

During training, we sample RoIs once at the first stage and keep the order of the RoIs for every stages to enable using adaptive feature alignment strategy. We use SGD as the optimizer with a weight decay of 0.0001 and a momentum of 0.9. The long edge and short edge of each image are resized to 1333 and 800, respectively, without changing the aspect ratio. The scale factors ($r$) to enlarge RoI regions are set to $2^{\frac{3}{3}}, 2^{\frac{2}{3}}, 2^{\frac{1}{3}}$ for stage $1, 2, 3$ respectively.

During inference, following HTC [9], at the last box head, the predicted boxes with confidence scores lower than $0.001$ are filtered out. Then, standard non-maximum suppression (NMS) (IoU threshold $= 0.5$) is applied to remove duplicated boxes.

### 4.2. Benchmarking Results

All the models are trained for 20 epochs with the learning rate decays at 16th and 19th epochs. For large models that cannot be fitted in the memory, we set the batch size to 8 and adjust the initial learning rate to 0.01.

**COCO** `val`: We conduct experiments on COCO `val` to show the improvements of our DSC over HTC. The experimental results summarized in Table 1 show that DSC achieves consistent and remarkable improvements (2.1 box AP and 1.5 mask AP) over HTC with different backbones *e.g.*, ResNet (R) and ResNeXt (X). Note that, DSC always achieves larger improvements under the more strict metric, *i.e.*, $AP^{75}$, showing high precision of our predictions. Fig. 5 shows the qualitative comparisons between HTC and DSC, taking X-101-32x4d FPN as the backbone.

**COCO** `test-dev`: We do a comprehensive comparison with state-of-the-art instance segmentation methods with strong network backbones on COCO `test-dev`. The results are summarized in Table 2, which show that DSC outperforms these state-of-the-art methods by a larger margin when using the same network backbones. Especially, HTC with deformable convolution (DCN) [18] and multi-scale training (ms train) is a very powerful cascade model for instance segmentation, which already achieved very high performance (50.8 box AP and 44.2 mask AP). Nevertheless, our method, DSC, with the same backbone and training strategy (DCN + ms train) still obtains a significant improvement (1.0 box AP and 1.3 mask AP) and achieves promising results (51.8 box AP and 45.5 mask AP) on COCO `test-dev`.

### 4.3. Ablation Study

In this section, we conduct ablation studies on COCO `val` to investigate the efficiency of our method, the contribution of each component we introduced for the cascade architecture. We use R-50 as the backbone and $1\times$ learning rate schedule for all ablation studies.

#### 4.3.1 Precision vs Inference Time

There is always a trade-off between precision and inference time for a method. Since our method has two versions, DSC and a fast version F-DSC. It is necessary to discuss this trade-off for them. HTC is taken as the baseline for reference.

As the results shown in Tab. 3, comparing to DSC, F-DSC achieves a comparable performance in terms of both Box AP and Mask AP, *i.e.*, only $0.3$ Box AP and $0.1$ Mask AP drops, while it is much faster than DSC, *i.e.*, reducing the inference time by 178ms. Comparing to HTC, F-DSC achieves significant improvements in both Box AP and Mask AP, *i.e.*, $2.2$ Box AP and $2.0$ Mask AP improvements. Moreover, the additional inference time ($18$ ms) is negligible. Since F-DSC performs excellently in both precision and inference time, we conduct the rest experiments based on F-DSC.

#### 4.3.2 Contribution of Each Cascade Component

We conduct an ablation study to verify the contribution of each component we introduce for our cascade architecture, including shape-guided RoIAlign, *i.e.*, explicit shape guidance (ExSG), the "+1" term in shape-guided RoIAlign (Plus1), the feature fusion operation, *i.e.*, implicit shape guidance (ImSG), and the adaptive feature alignment (AFA) strategy. We also compare a baseline which recomputes the mask features for the updated RoIs of each stage (ReComp) instead of using the AFA strategy. Since mRPN only provides the initial shape guidance for the first stage of our cascade architecture, we exclude it from the ablation study.

Table 4 summarizes the result of ablation study. We observe that excluding either ExSG or ImSG from F-DSC leads to performance degradation, showing that both explicit and implicit shape guidances provide positive feedback for the cascade architecture. we find that removing the "+1" term from ExSG leads to a drop of 0.2 Box AP and 0.2 Mask AP. Regarding that the improvement brought by ExSG is 0.3 Box AP/0.4 Mask AP, the "+1" term is important. We also observe that, without the AFA strategy, the performance has a noticeable drop (0.5 Box AP and 0.5 Mask AP). This result shows that the misaligned mask features impose negative effect on the implicit shape guidance and the adaptive feature alignment method is effective to address this problem. Recomputing the mask features for the updated RoIs of each stage can solve this misalignment

| Method | Backbone | $AP_b$ | $AP_b^{50}$ | $AP_b^{75}$ | $AP_m$ | $AP_m^{50}$ | $AP_m^{75}$ |
|---|---|---|---|---|---|---|---|
| HTC | R-50 FPN | 43.3 | 62.2 | 47.1 | 38.3 | 59.3 | 41.4 |
| DSC | R-50 FPN | 45.8(+2.5) | 63.4 (+1.2) | 49.8(+2.7) | 40.2 (+1.9) | 61.0(+1.7) | 43.5(+2.1) |
| HTC | R-101 FPN | 44.8 | 63.3 | 48.8 | 39.6 | 61.0 | 42.8 |
| DSC | R-101 FPN | 46.6(+1.8) | 64.5 (+1.2) | 50.8(+2.0) | 40.7 (+1.1) | 62.0 (+1.0) | 44.1(+1.3) |
| HTC | X-101-32x4d FPN | 46.1 | 65.3 | 50.1 | 40.5 | 62.5 | 43.7 |
| DSC | X-101-32x4d FPN | 48.0(+1.9) | 65.9(+0.6) | 52.2(+2.1) | 42.0(+1.5) | 63.7(+1.2) | 45.6(+1.9) |

Table 1: Comparison with HTC on COCO `val`.

| Method | Backbone | Epoch | $AP_b$ | $AP_b^{50}$ | $AP_b^{75}$ | $AP_m$ | $AP_m^{50}$ | $AP_m^{75}$ |
|---|---|---|---|---|---|---|---|---|
| *One-stage:* | | | | | | | | |
| BlendMask (ms train) [8] | R-50+FPN | 36 | - | - | - | 37.0 | 58.9 | 39.7 |
| SOLOv1 (ms train) [49] | R-50+FPN | 72 | - | - | - | 36.8 | 58.6 | 39.0 |
| SOLOv2 (ms train) [50] | R-50+FPN | 72 | - | - | - | 38.8 | 59.9 | 41.7 |
| CondInst (ms train) [47] | R-50+FPN | 36 | - | - | - | 38.8 | 60.4 | 41.5 |
| *Two-stage:* | | | | | | | | |
| FCIS++ [28] | R-101 | - | - | - | - | 33.6 | 54.5 | - |
| MaskLab+ [11] | R-101(JET) | - | - | - | - | 38.1 | 61.1 | 40.4 |
| PANet [34] | R-50+FPN | 24 | - | - | - | 36.6 | 58.0 | 39.3 |
| D2Det [7] | R-101+FPN | 24 | - | - | - | 40.2 | 61.5 | 43.7 |
| Mask R-CNN [23] | R-101+FPN | 24 | 41.6 | 62.5 | 45.3 | 37.4 | 59.5 | 40.0 |
| MS R-CNN [24] | R-101+FPN | 24 | 41.6 | 62.3 | 46.2 | 38.3 | 58.5 | 41.5 |
| *Cascade:* | | | | | | | | |
| Cascade Mask R-CNN[6] | R-50+FPN | 20 | 42.8 | 61.6 | 46.5 | 37.0 | 58.6 | 39.9 |
| HTC [9] | R-50+FPN | 20 | 43.6 | 62.6 | 47.4 | 38.5 | 60.1 | 41.7 |
| DSC (ours) | R-50+FPN | 20 | 46.0 | 63.9 | 50.1 | 40.5 | 61.8 | 44.1 |
| HTC [9] | R-101+FPN | 20 | 45.1 | 64.2 | 49.1 | 39.8 | 61.6 | 43.1 |
| DSC(ours) | R-101+FPN | 20 | 46.7 | 64.7 | 50.9 | 40.9 | 62.5 | 44.5 |
| HTC [9] | X-101-32x4d+FPN | 20 | 46.4 | 65.8 | 50.4 | 41.0 | 63.2 | 44.4 |
| DSC(ours) | X-101-32x4d+FPN | 20 | 48.1 | 66.3 | 52.4 | 42.2 | 64.1 | 45.8 |
| HTC (ms train) [9] | X-101-64x4d+DCN+FPN | 20 | 50.8 | 70.3 | 55.2 | 44.2 | 67.8 | 48.1 |
| DSC (ms train) (ours) | X-101-64x4d+DCN+FPN | 20 | **51.8** | **70.5** | **56.7** | **45.5** | **68.4** | **49.7** |

Table 2: Comparison with state-of-the-art methods on COCO `test-dev`.

| Methods | $AP_b$ | $AP_m$ | Inference Time |
|---|---|---|---|
| HTC | 42.3 | 37.4 | 238ms |
| DSC | 44.8(+2.5) | 39.5(+2.1) | 434ms(+196ms) |
| F-DSC | 44.5(+2.2) | 39.4 (+2.0) | 256ms(+18ms) |

Table 3: Comparison among DSC, F-DSC and HTC

problem and further improves F-DSC by 0.4 Box AP and 0.4 Mask AP, but takes extra 101ms.

### 4.4. Quantitative Results on huddle Instances

**Huddled Instance Data Collection.** To validate the advantage of DSC on detecting and segmenting huddled instances, we do comparisons between DSC and HTC on subsets of COCO 2017 `val` with different proportions of huddled instances. These subsets are collected according to two controllable thresholds. The first one is an intersection-over-union threshold $T_O$, to determine whether an instance is huddled, *i.e.*, a huddled instance should have a large overlap ($> T_O$) with other instances; The second one is a proportion threshold $T_P$, to determine whether an image contains a large proportion of huddled instances, *i.e.*, the number of huddled instances over the number of all instances in the image should be $> T_P$. We collect the subsets by varying $T_O$ and $T_P$ from 0.0 to 0.4 with an interval of 0.1.

| Method | $AP_b$ | $AP_m$ |
|---|---|---|
| F-DSC | 44.5 | 39.5 |
| F-DSC - ExSG | 44.2(-0.3) | 39.1(-0.4) |
| F-DSC - ImSG | 43.4(-1.1) | 38.8(-0.7) |
| F-DSC - Plus1 | 44.3(-0.2) | 39.3(-0.2) |
| F-DSC - AFA | 44.0(-0.5) | 39.0(-0.5) |
| F-DSC - AFA + ReComp | 44.9(+0.4) | 39.9(+0.4) |

Table 4: Ablation study to verify the contribution of each component we introduce for our cascade architecture. The symbols "+" and "-" mean including and excluding a component into and from F-DSC, respectively.

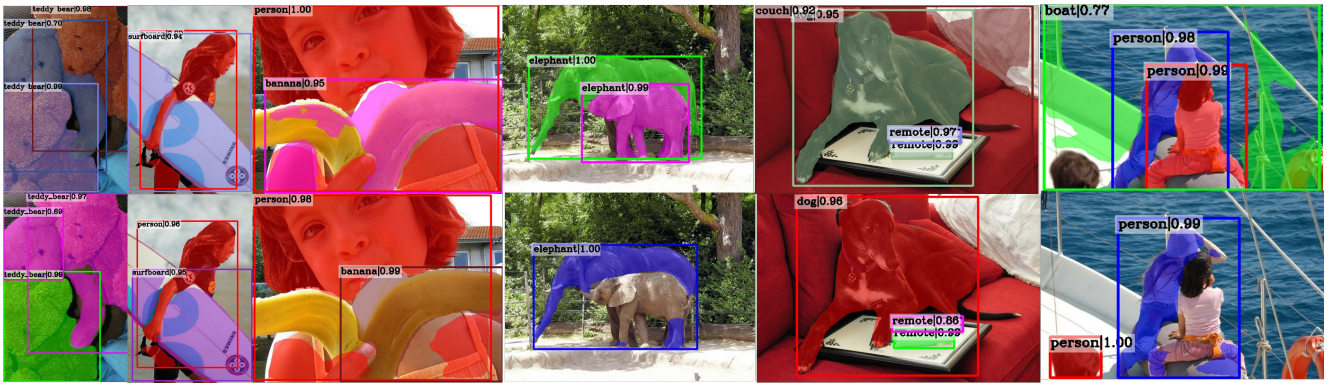


Figure 5: Qualitative comparison between DSC (top) and HTC (bottom) on COCO `val`. All selected images contain huddled instances. HTC is unable to predict object boxes precisely (the left three) or to recognize objects correctly (the right three). DSC successfully recognizes all the objects and segments them out.

**Evaluation Metric** The standard COCO-style evaluation metric is mean Average Precision (AP) over classes. But, in a subset, some classes may only have very a few instances or even no instance, making the standard AP heavily unstable and biased. Therefore, we compute **AP over all instances**, denoted by APoI, as the metric to evaluate the results on the subsets instead.

**Results** We use $1\times$ schedule to train an F-DSC and an HTC. The backbone is R-50 FPN. We report the improvements in terms of box APoI and mask APoI of DSC over HTC in Table 5. Note that, when $T_O = -1.0$, the selected subset is the original COCO 2017 `val` set, since all instances are determined as huddled instances. F-DSC outperforms HTC by 2.4 box APoI and 2.1 mask APoI respectively on the original set. Note that, the improvements in terms of APoI are similar to the improvements in terms of AP reported in Table 1, which shows APoI is a reasonable metric. With the increase of $T_O$ and $T_P$, we observe the improvements of DSG over HTC become larger, achieving **4.2** box APoI and **4.7** mask APoI on subsets with high proportions of heavily overlapped instances. These results evidence the advantage of DSC on detecting and segmenting huddled instances.

## 5. Limitation

We investigate the failure cases of our proposed method to study the limitation of our method. Since our method is shape-guided, it can be imagined that the final result quality of our method relies on informative initial mask predictions. So, bad initial mask predictions might worsen final results. To verify this hypothesis, we investigate the comparison between DSC and HTC case-by-case and find that 1) DSC performs worse on only a few cases ($\approx 10\%$), 2) the qualities of the initial mask predictions of these cases are low ($\approx 0.3$ mask mean IoU). To address this limitation, we can explore the direction of estimating mask IoUs, e.g. Mask Scoring, and re-weighting shape guidances according to the estimated IoUs.

| $T_O$ \ $T_P$ | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| -1.0 | 2.4 / 2.1 | - | - | - | - |
| 0.0 | 2.5 / 2.0 | 2.5 / 2.0 | 2.6 / 2.2 | 2.7 / 2.4 | 2.5 / 2.4 |
| 0.1 | 2.5 / 2.0 | 2.4 / 2.1 | 2.6 / 2.3 | 2.8 / 2.5 | 2.8 / 2.8 |
| 0.2 | 2.5 / 2.1 | 2.5 / 2.2 | 2.7 / 2.6 | 2.9 / 2.9 | 3.0 / 3.3 |
| 0.3 | 2.4 / 2.0 | 2.6 / 2.4 | 3.0 / 2.9 | 3.7 / 3.6 | 3.6 / 3.9 |
| 0.4 | 2.5 / 2.1 | 2.9 / 2.6 | 3.2 / 3.2 | 3.8 / 3.9 | **4.2** / **4.7** |

Table 5: Improvements of F-DSC over HTC on subsets of COCO 2017 `val` with different proportions of huddled instances. In each table cell, we report the box APoI improvement / the mask APoI improvement on the subset selected by using the two thresholds $T_O$ and $T_P$.

## 6. Conclusion

In this paper, we proposed Deeply Shape-guided Cascade (DSC) for instance segmentation, which iteratively makes use of the shape guidances extracted from mask segmentation at previous stage to improve bounding box detection at current stage. Then, the improved bounding box detection results can lead to more precise mask segmentation at current stage. This forms a positive feedback loop between bounding box detection and mask segmentation across multiple stages in the cascade, establishing a bi-directional relationship between the two tasks. Results on the COCO benchmark showed that DSC outperforms the state-of-the-art instance segmentation cascade, HTC, by a large margin. Particularly, DSC achieved significant improvements over HTC on segmenting huddled instances.

**Acknowledgement** This work was supported by NSFC 61672336, NSFC U19B2035, Shanghai Municipal Science and Technology Major Project 2021SHZDZX0102, and ONR N00014-20-1-2206.

## References


[1] Pablo Andrés Arbeláez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marqués, and Jitendra Malik. Multiscale combinatorial grouping. In *Proc. CVPR*, pages 328–335, 2014. 1

[2] Anurag Arnab and Philip H. S. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Proc. CVPR*, pages 879–888, 2017. 2

[3] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proc. CVPR*, pages 2858–2866, 2017. 2

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: real-time instance segmentation. In *Proc. ICCV*, 2019. 2

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *Proc. CVPR*, pages 6154–6162, 2018. 1, 3

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. 1, 3, 7

[7] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11482–11491. IEEE, 2020. 3, 7

[8] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *Porc. CVPR*, pages 8570–8578, 2020. 7

[9] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *Proc. CVPR*, pages 4974–4983, 2019. 1, 2, 3, 6, 7

[10] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[11] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proc. CVPR*, pages 4013–4022, 2018. 3, 7

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 1

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

[14] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *Proc. CVPR*, pages 3470–3478, 2015. 1

[15] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask r-cnn. In *Proc. ECCV*, 2020. 3

[16] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *Proc. ECCV*, pages 534–549, 2016. 2

[17] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *Proc. NIPS*, pages 379–387, 2016. 1

[18] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proc. ICCV*, pages 764–773, 2017. 6

[19] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proc. CVPR*, pages 2155–2162, 2014. 1

[20] Ross B. Girshick. Fast R-CNN. In *Proc. ICCV*, pages 1440–1448, 2015. 1, 3

[21] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *Proc. ECCV*, pages 297–312, 2014. 1

[22] Adam W. Harley, Konstantinos G. Derpanis, and Iasonas Kokkinos. Segmentation-aware convolutional networks using local attention masks. In *Proc. ICCV*, pages 5048–5057, 2017. 1, 2

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *Proc. ICCV*, pages 2980–2988, 2017. 3, 7

[24] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring R-CNN. In *Proc. CVPR*, pages 6409–6418, 2019. 3, 7

[25] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: From edges to instances with multicut. In *Proc. CVPR*, pages 7322–7331, 2017. 2

[26] Chen-Yu Lee, Saining Xie, Patrick W. Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proc. AISTATS*, 2015. 1

[27] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5325–5334. IEEE Computer Society, 2015. 3

[28] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proc. CVPR*, pages 4438–4446, 2017. 3, 7

[29] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9128–9137. IEEE, 2020. 2

[30] Xiaodan Liang, Liang Lin, Yunchao Wei, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Proposal-free network

for instance-level object segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2978–2991, 2018. 2

[31] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, pages 936–944, 2017. 1

[32] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[33] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. SGN: sequential grouping networks for instance segmentation. In *Proc. ICCV*, pages 3516–3524, 2017. 2

[34] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proc. CVPR*, pages 8759–8768, 2018. 3, 7

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *Proc. ECCV*, pages 21–37, 2016. 1

[36] AudeO liva and Antonio Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520 – 527, 2007. 4

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, pages 3431–3440, 2015. 1

[38] Ping Luo, Guangrun Wang, Liang Lin, and Xiaogang Wang. Deep dual learning for semantic image segmentation. In *Proc. ICCV*, pages 2737–2745, 2017. 1

[39] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proc. CVPR*, pages 8837–8845, 2019. 2

[40] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Proc. NIPS*, pages 2277–2287, 2017. 2

[41] Wanli Ouyang, Kun Wang, Xin Zhu, and Xiaogang Wang. Chained cascade network for object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1956–1964. IEEE Computer Society, 2017. 3

[42] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8530–8539. IEEE, 2020. 2

[43] Pedro H. O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Proc. NIPS*, pages 1990–1998, 2015. 1, 2

[44] Pedro Oliveira Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *Proc. ECCV*, pages 75–91, 2016. 1, 2

[45] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, pages 779–788, 2016. 1

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, 2015. 1, 3

[47] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2020. 2, 7

[48] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: segmenting objects by locations. *CoRR*, abs/1912.04488, 2019. 2

[49] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: segmenting objects by locations. In *Proc. ECCV*, volume 12363 of *Lecture Notes in Computer Science*, pages 649–665, 2020. 7

[50] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *Proc. NeurIPS*, 2020. 7

[51] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic, faster and stronger. *CoRR*, abs/2003.10152, 2020. 2

[52] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Bridging category-level and instance-level semantic image segmentation. *ArXiv*, 1605.06885, 2016. 2

[53] 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. 2

[54] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. ICCV*, pages 1395–1403, 2015. 1

[55] Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 1305–1316, 2019. 2

[56] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z. Li. CRAFT objects from images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 6043–6051. IEEE Computer Society, 2016. 3

[57] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Proc. CVPR*, pages 669–677, 2016. 2

[58] Ziyu Zhang, Alexander G. Schwing, Sanja Fidler, and Raquel Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *Proc. ICCV*, pages 2614–2622, 2015. 2

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. CVPR*, pages 6230–6239, 2017. 1

[60] Xiangyun Zhao, Shuang Liang, and Yichen Wei. Pseudo mask augmented object detection. In *Proc. CVPR*, pages 4061–4070, 2018. 1

[61] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *Proc. ICCV*, pages 1529–1537, 2015. 1

[62] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *Proc. CVPR*, pages 850–859, 2019. 2