CenterMask : Real-Time Anchor-Free Instance Segmentation

Youngwan Lee and Jongyoul Park*
Electronics and Telecommunications Research Institute (ETRI), South Korea
`{yw.lee, jongyoul}@etri.re.kr`

Abstract

We propose a simple yet efficient anchor-free instance segmentation, called CenterMask, that adds a novel spatial attention-guided mask (SAG-Mask) branch to anchor-free one stage object detector (FCOS [33]) in the same vein with Mask R-CNN [9]. Plugged into the FCOS object detector, the SAG-Mask branch predicts a segmentation mask on each detected box with the spatial attention map that helps to focus on informative pixels and suppress noise. We also present an improved backbone networks, VoVNetV2, with two effective strategies: (1) residual connection for alleviating the optimization problem of larger VoVNet [19] and (2) effective Squeeze-Excitation (eSE) dealing with the channel information loss problem of original SE. With SAG-Mask and VoVNetV2, we deign CenterMask and CenterMask-Lite that are targeted each to large and small models, respectively. Using the same ResNet-101-FPN backbone, CenterMask achieves 38.3%, surpassing all previous state-of-the-art methods while at a much faster speed. CenterMask-Lite also outperforms the state-of-the-art by large margins at over 35fps on Titan Xp. We hope that CenterMask and VoVNetV2 can serve as a solid baseline of real-time instance segmentation and backbone network for various vision tasks, respectively. The Code is available at `https://github.com/youngwanLEE/CenterMask`.

1. Introduction

Recently, instance segmentation has made great progress beyond object detection. The most representative method, Mask R-CNN [9], extended on object detection (e.g., Faster R-CNN [30]), has dominated COCO [23] benchmarks since instance segmentation can be easily solved by detecting objects and then predicting pixels on each box. However, even if there have been many works [15, 2, 3, 20, 24] for improving the Mask R-CNN [9], few works exist for considering the speed of the instance segmentation. Although YOLACT [1] is the first real-time one-stage instance segmentation due to its parallel structure and extremely lightweight assembly process, the accuracy gap from Mask R-CNN [9] is still significant. Thus, we aim to bridge the gap by improving both accuracy and speed.

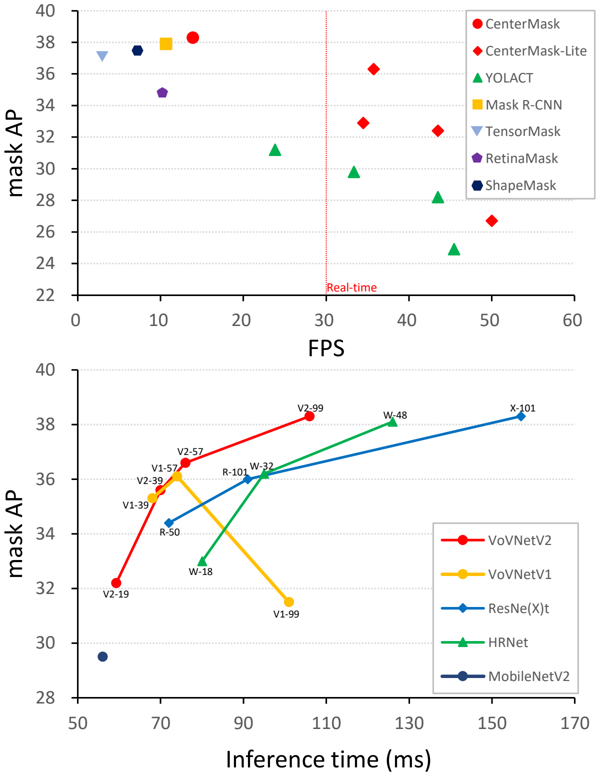


Figure 1: **Accuracy-speed Tradeoff.** across various instance segmentation models (top) and backbone networks (bottom) on COCO. The inference speed of CenterMask & CenterMask-Lite is reported on the same GPU (V100/Xp) with their counterparts. Note that all backbone networks in the bottom are compared under the proposed CenterMask. Please refer to section 3.2, Table 3 and Table 5 for details.

While Mask R-CNN [9] is based on a two-stage object detector (e.g., Faster R-CNN) that first generates box proposals and then predicts box location and classification, YOLACT [1] is built on one-stage detector (RetinaNet [22])

*Corresponding author.

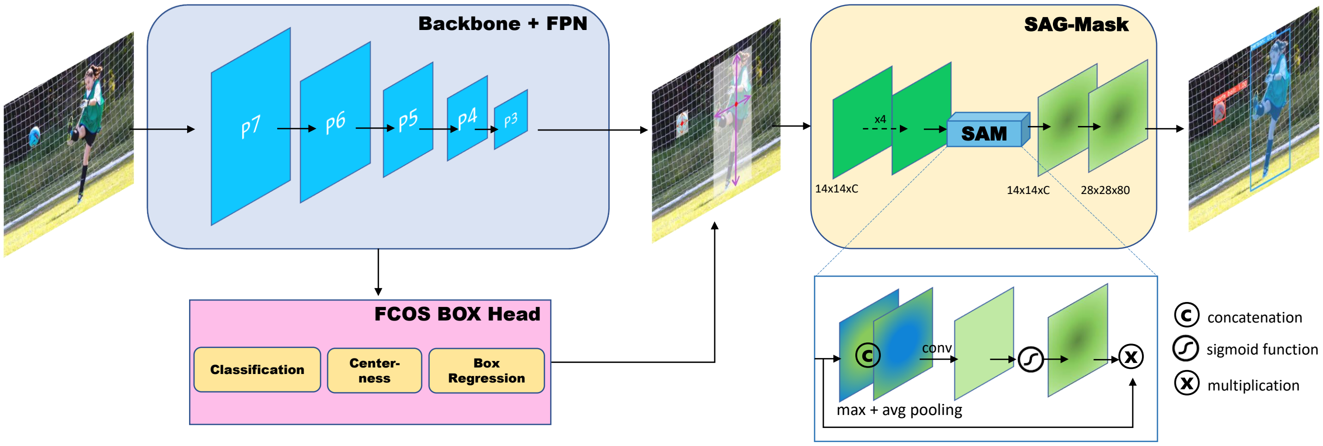


Figure 2: **Architecture of CenterMask.** where P3 (stride of 2^3) to P7 (stride of 2^7) denote the feature map in feature pyramid of backbone network. Using the features from the backbone, FCOS [33] predicts bounding boxes. Spatial Attention-Guided Mask (SAG-Mask) predicts segmentation mask inside of the each detected box with Spaital Attention Module (SAM) helping to focus on the informative pixels but also suppress the noise.

that directly predicts boxes without proposal step. However, these object detectors rely heavily on pre-define anchors, which are sensitive to hyper-parameters (e.g., input size, aspect ratio, scales, etc.) and different datasets. Besides, since they densely place anchor boxes for higher recall rate, the excessively many anchor boxes cause the imbalance of positive/negative samples and higher computation/memory cost. To cope with these drawbacks of anchor boxes, recently, many works [18, 6, 38, 39, 33, 38] tend to escape from the anchor boxes toward *anchor-free* by using corner/center points, which leads to more computation-efficient and better performance compared to anchor box based detectors.

Therefore, we design a simple yet efficient anchor-free one stage instance segmentation called *CenterMask* that adds a novel spatial attention-guided mask branch to the more efficient one-stage anchor-free object detector (FCOS [33]) in the same way with Mask R-CNN [9]. Figure 2 shows the overview of our CenterMask. Plugged into the FCOS [33] object detector, our spatial attention-guided mask (SAG-Mask) branch takes the predicted boxes from the FCOS [33] detector to predict segmentation masks on each Region of Interest (RoI). The spatial attention module (SAM) in the SAG-Mask helps the mask branch to focus on meaningful pixels and suppressing uninformative ones.

When extracting features on each RoI for mask prediction, each RoI pooling should be assigned considering the RoI scales. Mask R-CNN [9] proposes a new assignment function, called RoIAlign, that does not consider the input scale. Thus, we design a scale-adaptive RoI assignment function that considers the input scale and is a more suitable one-stage object detector. We also propose a more effective backbone network *VoVNetV2* based on VoVNet [19] that shows better performance and faster speed than ResNet [10] and DenseNet [14] due to its One-shot Aggregation (OSA). In Figure 1 (bottom), We found that stacking the OSA modules in VoVNet makes the performance degradation (e.g., VoVNetV1-99). We see this phenomenon as the motivation of ResNet [10] because the backpropagation of gradient is disturbed. Thus, we add the residual connection [10] into each OSA module to ease optimization, which makes the VoVNet deeper and in turn, boosts the performance.

In the Squeeze-Excitation (SE) [13] channel attention module, it was found that the fully connected layers reduce the channel size, thereby reducing computational burden and unexpectedly causing channel information loss. Thus, we re-design the SE module as *effective* SE (eSE) replacing the two FC layers with one FC layer maintaining channel dimension, which prevents the information loss and in turn, improves the performance. With residual connection and eSE modules, We propose VoVNetV2 on various scales; from lightweight VoVNetV2-19, base VoVNetV2-39/57 and large model VoVNetV2-99 that are correspond with MobileNet-V2 [11], ResNet-50/101 [10] & HRNet-W18/32 [32], and ResNeXt-32x8d [36].

With SAG-Mask and VoVNetV2, we design CenterMask and CenterMask-Lite that are targeted each to large and small models, respectively. The Extensive experiments demonstrate the effectiveness of CenterMask & CenterMask-Lite and VoVNetV2. Using the same ResNet-101 backbone [10], CenterMask outperforms all previous state-of-the-art single models on the COCO [23] instance and detection tasks while at a much faster speed. CenterMask-Lite with VoVNetV2-39 bakcbone also achieves 33.4% mask AP / 38.0% box AP, outperforming the state-of-the-art real-time instance segmentation YOLACT [1] by 2.6 / 7.0 AP gain, respectively, at over 35fps on Titan Xp.

2. CenterMask

In this section, first, we review the anchor-free object detector, FCOS [33], which is a fundamental object detection part of our CenterMask. Next, we demonstrate the architecture of the CenterMask and describe how the proposed spatial attention-guided mask branch (SAG-Mask) is designed to plug into the FCOS [33] detector. Finally, a more effective backbone network, VoVNetV2, is proposed to boost the performance of CenterMask in terms of accuracy and speed.

2.1. FCOS

FCOS [33] is an anchor-free and proposal-free object detection in a per-pixel prediction manner as like FCN [26]. Almost state-of-the-art object detectors such as Faster R-CNN [30], YOLO [29], and RetinaNet [22] use the concept of the pre-defined anchor box which needs elaborate parameter tunning and complex calculation associated with box IoU in training. Without the anchor-box, the FCOS [33] directly predicts a 4D vector plus a class label at each spatial location on a level of feature maps. As shown in Figure 2, the 4D vector embeds the relative offsets from the four sides of a bounding box to the location (e.g., left, right, top and bottom). In addition, FCOS [33] introduces the centerness branch to predict the deviation of a pixel to the center of its corresponding bounding box, which improves the detection performance. Avoiding complex computation of anchor-boxes, FCOS [33] reduces memory/computation cost but also outperforms the anchor box based object detectors. Because of the efficiency and good performance of the FCOS [33], we design the proposed CenterMask built upon the FCOS [33] object detector.

2.2. Architecture

Figure 2 shows overall architecture of the CenterMask. CenterMask consists of three-part:(1) backbone for feature extraction, (2) FCOS [33] detection head, and (3) mask head. The procedure of masking objects is composed of detecting objects from the FCOS [33] box head and then predicting segmentation masks inside the cropped regions in a per-pixel manner.

2.3. Adaptive RoI Assignment Function

After object proposals are predicted in the FCOS [33] box head, CenterMask predicts segmentation masks using the predicted box regions in the same vein as Mask R-CNN. As the RoIs are predicted from different levels of feature maps in Feature Pyramid Network (FPN [21]), RoI Align [9] that extracts features should be assigned at different scales of feature maps with respect to RoI scales. Specifically, an RoI with a large scale has to be assigned to a higher feature level and vice versa. Mask R-CNN [9] based two-stage detector uses Equation 1 in FPN [21] to determine which feature map (P_k) to be assigned.

$$k = \lfloor k_0 + \log_2 \sqrt{wh}/224 \rfloor, \qquad (1)$$

where k_0 is 4 and w, h are the width and height of the each RoI. However, Equation 1 is not suitable for CenterMask based one-stage detector because of two reasons. First, Equation 1 is tuned to two-stage detectors (e.g.,FPN [21]) that use different feature levels compared to one-stage detectors (e.g, FCOS [33], RetinaNet [22]). Specifically, two-stage detectors use feature levels of P2 (stride of 2^2) to P5 (2^5) while one-stage detectors use from P3 (2^3) to P7 (2^7) that is larger receptive fields with lower-resolution. Besides, the canonical ImageNet pretraining size 224 in Equation 1 is hard-coded and not adaptive to feature scale variation. For example, when the input dimension is 1024×1024 and the area of an RoI is 224^2, the RoI is assigned to relative higher feature P4 despite its small size of the area with respect to input dimension, which results in reducing small object AP. Therefore, we define Equation 2 as a new RoI assignment function suited for CenterMask based one-stage detectors.

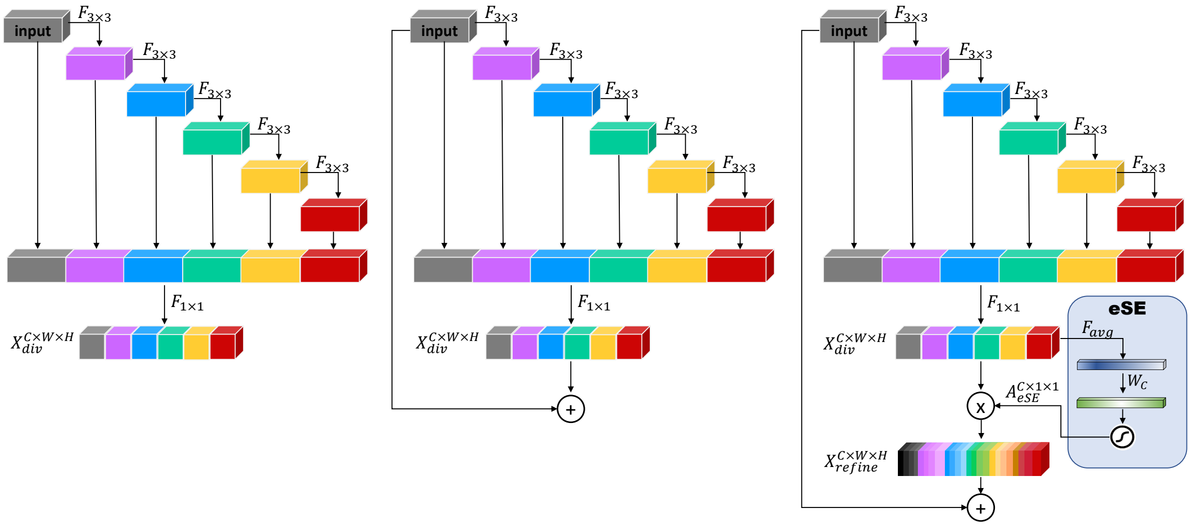
$$k = \lceil k_{\max} - \log_2 A_{input}/A_{RoI} \rceil, \qquad (2)$$

where $k_{\max}$ is the last level (e.g., 7) of feature map in backbone and A_{input}, A_{RoI} are area of input image and the RoI, respectively. Without the canonical size 224 in Equation 1, Equation 2 adaptively assign RoI pooling scale by the ratio of input/RoI area. If k is lower than minimum level (e.g., P3), k is clamped to the minimum level. Specifically, if the area of an RoI is bigger than half of the input area, the RoI is assigned to the highest feature level(e.g., P_7). Inversely, while Equation 1 assigns P_4 to the RoI with 224^2, Equation 2 determine $k_{\max}$ - 5 level which maybe minimum feature level for area of the RoI that is about ×20 smaller than input size. We can find that the proposed RoI assignment method improves the small object AP than Equation 1 because of its adaptive and scale-aware assignment strategy in Table 2. From an ablation study, we set $k_{\max}$ to P5 and $k_{\min}$ to P3.

2.4. Spatial Attention-Guided Mask

Recently, attention methods [13, 34, 40, 28] have been widely applied to object detections because it helps to focus on important features, but also suppress unnecessary ones. In particular, channel attention [13, 12] emphasizes 'what' to focus across channels of feature maps while spaital attention [34, 4] focuses 'where' is an informative regions. Inspired by the spatial attention mechanism, we adopt a spatial attention module to guide the mask head for spotlighting meaningful pixels and repressing uninformative ones.

Thus, we design a spatial attention-guided mask (SAG-Mask), as shown in Figure 2. Once features inside the predicted RoIs are extracted by RoI Align [9] with 14×14 resolution, those features are fed into four `conv` layers and



(a) OSA module in VoVNet **(b) OSA + identity mapping** **(c) OSA + identity mapping + eSE**

Figure 3: Comparison of OSA modules. $F_{1\times1}, F_{3\times3}$ denote $1\times1, 3\times3$ conv layer respectively, F_{avg} is global average pooling, W_C is fully-connected layer, A_{eSE} is channel attention map, $\otimes$ indicates element-wise multiplication and $\oplus$ denotes element-wise addition.

spatial attention module (SAM) sequentially. To exploit the spatial attention map $\mathrm{A}_{sag}(\mathrm{X}_i) \in \mathbb{R}^{1\times W\times H}$ as a feature descriptor given input feature map $\mathrm{X}_i \in \mathbb{R}^{C\times W\times H}$, the SAM first generates pooled features P_{avg}, $\mathrm{P}_{max} \in \mathbb{R}^{1\times W\times H}$ by both average and max pooling operations respectively along the channel axis and aggregates them via concatenation. Then it is followed by a 3×3 conv layer and normalized by the sigmoid function. The computation process is summarized as follow:

$$\mathrm{A}_{sag}(\mathrm{X}_i) = \sigma(\mathcal{F}_{3\times3}(\mathrm{P}_{max} \circ \mathrm{P}_{avg})), \tag{3}$$

where σ denotes the sigmoid function, $\mathcal{F}_{3\times3}$ is 3×3 conv layer and $\circ$ represents concatenate operation. Finally, the attention guided feature map $\mathrm{X}_{sag} \in \mathbb{R}^{C\times W\times H}$ is computed as:

$$\mathrm{X}_{sag} = \mathrm{A}_{sag}(\mathrm{X}_i) \otimes \mathrm{X}_i, \tag{4}$$

where $\otimes$ denotes element-wise multiplication. After then, a 2×2 deconv upsamples the spatially attended feature map to 28×28 resolution. Lastly, a 1×1 conv is applied for predicting class-specific masks.

2.5. VoVNetV2 backbone

In this section, we propose more effective backbone networks, *VoVNetV2*, for further boosting the performance of CenterMask. VoVNetV2 is improved from VoVNet [19] by adding residual connection [10] and the proposed effective Squeeze-and-Excitation (eSE) attention module to the VoVNet. VoVNet is a computation and energy efficient backbone network that can efficiently present diversified feature representation because of One-Shot Aggregation (OSA) modules. As shown in Figure 3(a) OSA module consists of consecutive conv layers and aggregates the subsequent feature maps at once, which can capture diverse receptive fields efficiently and in turn outperforms DenseNet and ResNet in terms of accuracy and speed.

Residual connection: Even with its efficient and diverse feature representation, VoVNet has a limitation in terms of optimization. As OSA modules are stacked (i.g., deeper) in VoVNet, we observe the accuracy of the deeper models is saturated or degradation. Specifically, Table 4 shows the accuracy of VoVNetV1-99 is lower than that of VoVNetV1-57. Based on the motivation of ResNet [10], We conjecture that stacking OSA modules make the backpropagation of gradient gradually hard due to the increase of transformation functions such as conv. Therefore, as shown in Figure 3(b), we also add the identity mapping [10] to OSA modules. Correctly, the input path is connected to the end of an OSA module that is able to backpropagate the gradients of every OSA module in an end-to-end manner on each stage like ResNet. Boosting the performance of VoVNet, the identity mapping also makes the VoVNet possible to enlarge its depth such as VoVNet-99.

Effective Squeeze-Excitation (eSE): For further boosting the performance of VoVNet, We also propose a channel attention module, effective Squeeze-Excitation (eSE), improving original SE [13] more *effectively*. Squeeze-Excitation (SE) [13], a representative channel attention method adopted in CNN architectures, explicitly models the interdependency between the channels of feature maps to enhance its representation. The SE module squeezes the spatial dependency by global average pooling to learn a channel specific descriptor and then two fully-connected

(FC) layers followed by a sigmoid function are used to rescale the input feature map to highlight only useful channels. In short, given input feature map $\mathrm{X}_i \in \mathbb{R}^{C \times W \times H}$, the channel attention map $\mathrm{A_{ch}(X_i)} \in \mathbb{R}^{C \times 1 \times 1}$ is computed as:

$$\mathrm{A}_{ch}(\mathrm{X}_i) = \sigma(\mathrm{W}_C(\delta(\mathrm{W}_{C/r}(\mathcal{F}_{gap}(\mathrm{X}_i))))), \tag{5}$$

where $\mathcal{F}_{gap}(\mathrm{X}) = \frac{1}{WH} \sum_{i,j=1}^{W,H} \mathrm{X}_{i,j}$ is channel-wise global average pooling, $\mathrm{W}_{C/r}, \mathrm{W}_C \in \mathbb{R}^{C \times 1 \times 1}$ are weights of two fully-connected layers, δ denotes ReLU non-linear operator and σ indicates sigmoid function.

However, it is assumed that the SE module has a limitation: channel information loss due to dimension reduction. For avoiding high model complexity burden, two FC layers of the SE module need to reduce channel dimension. Specifically, while the first FC layer reduces input feature channels C to C/r using reduction ratio r, the second FC layer expands the reduced channels to original channel size C. As a result, this channel dimension reduction causes channel information loss.

Therefore, we propose effective SE (eSE) that uses only one FC layer with C channels instead of two FCs without channel dimension reduction, which rather maintains channel information and in turn improves performance. the eSE process is defined as:

$$\mathrm{A}_{eSE}(\mathrm{X}_{div}) = \sigma(\mathrm{W}_C(\mathcal{F}_{gap}(\mathrm{X}_{div}))), \tag{6}$$

$$\mathrm{X}_{refine} = \mathrm{A}_{eSE}(\mathrm{X}_{div}) \otimes \mathrm{X}_{div}, \tag{7}$$

where $\mathrm{X}_{div} \in \mathbb{R}^{C \times W \times H}$ is the diversified feature map computed by 1×1 conv in OSA module. As a channel attentive feature descriptor, the $\mathrm{A}_{eSE} \in \mathbb{R}^{C \times 1 \times 1}$ is applied to the diversified feature map X_{div} to make the diversified feature more informative. Finally, when using the residual connection, the input feature map is element-wise added to the refined feature map X_{refine}. The details of How the eSE module is plugged into the OSA module are shown in Figure 3(c).

2.6. Implementation details

Since CenterMask is built on FCOS [33] object detector, we follow hyper-parameters of the FCOS [33] except for positive score threshold 0.03 instead of 0.05 Since FCOS [33] does not generate positive RoI samples well in initial training time. While using FPN levels 3 through 7 with 256 channels in the detection step, we use P3 $\sim$ P7 in the masking step, as mentioned in 2.3. We also use mask scoring [15] that recalibrates classification score considering predicted mask quality (e.g., mask IoU) in Mask R-CNN.

CenterMask-Lite: To achieve real-time processing, we try to make the proposed CenterMask lightweight. We downsize three parts: backbone, box head, and mask head. In the backbone, first, we reduce the channels C of FPN from 256 to 128, which can decrease the output of 3×3 conv in FPN but also input dimension of box and mask head. And then, we replace the backbone network with more lightweight VoVNetV2-19 that has 4 OSA modules on each stage comprised of 3 conv layers instead of 5 as in VoVNetv2-39/57. In the box head, there are four 3×3 conv layers with 256 channels on each classification and box branch where the centerness branch is shared with the box branch. We reduce the number of conv layer from 4 to 2 with 128 channels. Lastly, in the mask head, we also reduce the number of conv layers and channels in the feature extractor and mask scoring part from (4, 256) to (2, 128), respectively.

Training: We set the number of detection boxes from the FCOS [33] to 100, and the highest-scoring boxes are fed into the SAG-mask branch for training mask branch. We use the same mask target as Mask R-CNN that is made by the intersection between an RoI and its associated ground-truth mask. During training time, we define a multi-task loss on each RoI as:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{center} + \mathcal{L}_{box} + \mathcal{L}_{mask}, \tag{8}$$

where the classification loss $\mathcal{L}_{cls}$, centerness loss $\mathcal{L}_{center}$, and box regression loss $\mathcal{L}_{box}$ are same as those in [33] and $\mathcal{L}_{mask}$ is the average binary cross-entropy loss identical as in [9]. Unless specified, the input image is resized to have 800 pixels [21] along the shorter side and their longer side less or equal to 1333. We train CenterMask by using Stochastic Gradient Descent (SGD) for 90K iterations ($\sim$12 epoch) with a mini-batch of 16 images and initial learning rate of 0.01 which is decreased by a factor of 10 at 60K and 80K iterations, respectively. We use a weight decay of 0.0001 and a momentum of 0.9, respectively. All backbone models are initialized by ImageNet pre-trained weights.

Inference: At test time, the FCOS detection part yields 50 high-score detection boxes, and then the mask branch uses them to predict segmentation masks on each RoI. CenterMask/CenterMask-Lite use a single scale of 800/600 pixels for the shorter side, respectively.

3. Experiments

In this section, we evaluate the effectiveness of CenterMask on COCO [23] benchmarks. All models are trained on the train2017 and val2017 are used for ablation studies. Final results are reported on test-dev for comparison with state-of-the-arts. We use $\mathrm{AP}^{\mathrm{mask}}$ as *mask* average precision AP (averaged over IoU thresholds), $\mathrm{AP_S}$, $\mathrm{AP_M}$, and $\mathrm{AP_L}$ (AP at different scale). We also denote *box* AP as $\mathrm{AP}^{\mathrm{box}}$. All ablation studies are conducted using CenterMask with ResNet-50-FPN. We report the inference time of models using one *thread* (1 batch size) on the same enviroment equipped with Titan Xp GPU, CUDA v10.0,

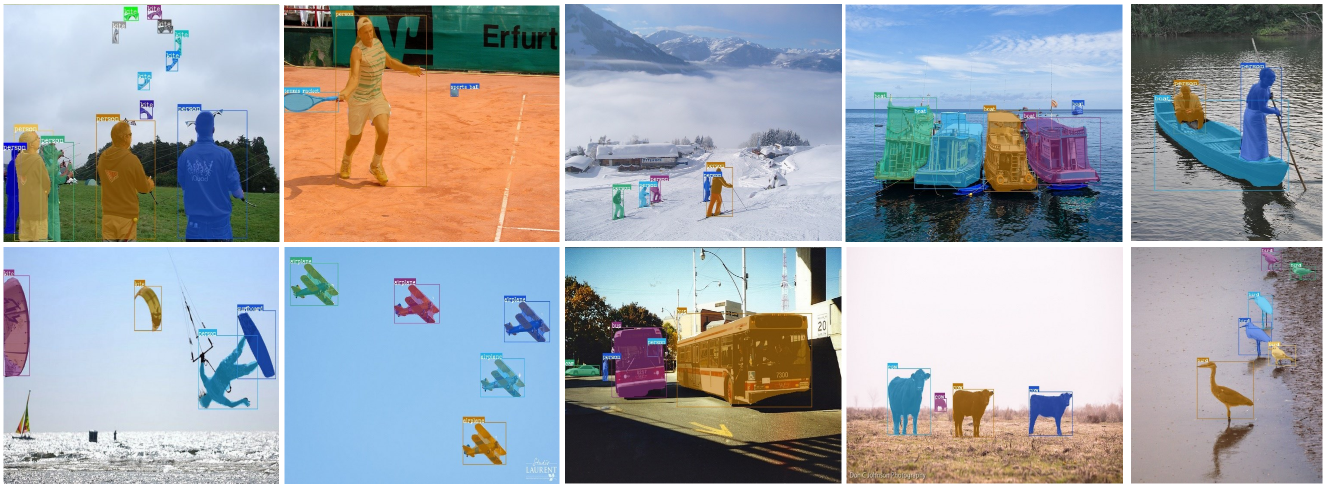


Figure 4: Results of CenterMask with VoVNetV2-99 on COCO `test-dev2017`.

Component	$\mathrm{AP}^{\mathrm{mask}}$	$\mathrm{AP}^{\mathrm{box}}$	Time (ms)
FCOS (baseline), *ours*	-	37.8	57
+ mask head (Eq. 1 [21])	33.4	38.3	67
+ mask head (Eq. 2, *ours*)	33.8	38.7	67
+ SAM	34.0	38.9	67
+ Mask scoring	34.7	38.8	72

Table 1: **Spatial Attention Guided Mask (SAG-Mask)** These models use ResNet-50 backbone. We note that the mask heads with Eq.1 is same as the mask branch of Mask R-CNN. SAM and Scoring denotes the proposed Spatial Attention Module and mask scoring [15].

cuDNN v7.3, and pytorch1.1. The Qualitative results of CenterMask are shown in Figure 4.

3.1. Ablation study

Scale-adaptive RoI assignment function: Comparing to Equation 1 [21], we validate the proposed Equation 2 in CenterMask. Table 1 shows that our scale-adaptive RoI assignment function considering the input scale improves by 0.4% $\mathrm{AP}^{\mathrm{mask}}$ over the counterpart. It means that Equation 2 regarding the ratio of input/RoI is more scale-adaptive than Equation 1.

Spatial Attention Guided Mask: Table 1 demonstrates the influence of each component in building Spatial Attention Guided Mask (SAG-Mask). The baseline, FCOS [33] object detector, starts from 37.8% $\mathrm{AP}^{\mathrm{box}}$ with the run time of 57 ms. Adding only naive mask head improves the box performance by 0.5% $\mathrm{AP}^{\mathrm{box}}$ and obtains 33.4% $\mathrm{AP}^{\mathrm{mask}}$. With the prementioned scale-adaptive RoI mapping strategy, our spatial attention module, *SAM*, makes the mask performance forward because the spatial attention module helps the mask predictor to focus on informative pixels but also suppress noise. It can also be seen that the detection performance is boosted when using SAM. We suggest that result from the SAM, the refined feature maps of mask head would also have a secondary effect on the detection branch that shares feature maps of the backbone.

Feature Level	$\mathrm{AP}^{\mathrm{mask}}$	$\mathrm{AP}^{\mathrm{box}}$
P3 $\sim$ P7	34.4	38.8
P3 $\sim$ P6	34.6	38.8
P3 $\sim$ P5	**34.6**	**38.9**
P3 $\sim$ P4	34.4	38.5

Table 2: Feature level ranges for RoIAlign [9] in CenterMmask. P3$\sim$P7 denotes the feature maps with output stride of $2^3 \sim 2^7$

The SAG-mask also deploys the mask scoring [15] that recalibrates the score regarding the predicted mask IoU. As a result, the mask scoring increases performance by 0.7% $\mathrm{AP}^{\mathrm{mask}}$. We note that the mask scoring cannot boost detection performance because the recalibrated mask score adjusts the ranks of mask results in the evaluation step, not refines the features of the mask head like the SAM. Besides, SAM rarely causes extra computation while the mask scoring leads to computation overhead (e.g., +5ms).

Feature selection. We also ablate which feature level range is suitable for our CenterMask based one-stage detector. Since FCOS [33] detector extract features from P3 $\sim$ P7, we start the same feature levels in the SAG-mask branch. As shown in Table 2, the performance of the P3 $\sim$ P7 range is not as good as other ranges. We speculate P7 feature map is too small to extract fine features for pixel-level prediction (e.g., 7 × 7). We observe that P3 $\sim$ P5 feature range achieves the best result, which means feature maps with a bigger resolution are advantageous for the mask prediction.

VoVNetV2: We extend VoVNet to VoVNetV2 by using

Backbone	Params.	AP^{mask}	$\text{AP}^{\text{mask}}_{S}$	$\text{AP}^{\text{mask}}_{M}$	$\text{AP}^{\text{mask}}_{L}$	AP^{box}	$\text{AP}^{\text{box}}_{S}$	$\text{AP}^{\text{box}}_{M}$	$\text{AP}^{\text{box}}_{L}$	Time (ms)
MobileNetV2 [31]	28.7M	29.5	12.0	31.4	43.8	32.6	17.8	35.2	43.2	**56**
VoVNetV2-19 [19]	37.6M	**32.2**	**14.1**	**34.8**	**48.1**	**35.9**	**20.8**	**39.2**	**47.6**	59
HRNetV2-W18 [32]	36.4M	33.0	14.3	34.7	49.9	36.7	20.7	39.4	49.3	80
ResNet-50 [10]	51.2M	34.7	15.5	37.6	51.5	38.8	22.4	42.5	51.1	72
VoVNetV1-39 [19]	49.0M	35.3	15.5	38.4	52.1	39.7	23.0	43.3	52.7	**68**
VoVNetV2-39	52.6M	**35.6**	**16.0**	**38.6**	**52.8**	**40.0**	**23.4**	**43.7**	**53.9**	70
HRNetV2-W32 [32]	56.2M	36.2	16.0	38.4	53.0	40.6	23.0	43.8	53.1	95
ResNet-101[10]	70.1M	36.0	16.5	39.2	54.4	40.7	23.4	44.3	54.7	91
VoVNetV1-57 [19]	63.0M	36.1	16.2	39.2	54.0	40.8	23.7	44.2	55.3	**74**
VoVNetV2-57	68.9M	**36.6**	**16.9**	**39.8**	**54.5**	**41.5**	**24.1**	**45.2**	**55.2**	76
HRNetV2-W48 [32]	92.3M	38.1	17.6	41.1	55.7	43.0	25.8	46.7	55.9	126
ResNeXt-101 [36]	114.3M	38.3	**18.4**	41.6	55.4	43.1	**26.1**	46.8	55.7	157
VoVNetV1-99 [19]	83.6M	31.5	13.5	33.5	46.5	35.3	19.7	38.1	46.6	**101**
VoVNetV2-99	96.9M	**38.3**	18.0	**41.8**	**56.0**	**43.5**	25.8	**47.8**	**57.3**	106

Table 3: **CenterMask with other backbones** on COCO `val2017`. Note that all mdoels are trained with a same manner (e.g., 12 epoch, 16 batch size, without train & test augmentation). The inference time is reported on same Titan Xp GPU.

Backbone	Params.	AP^{mask}	AP^{box}	Time (ms)
VoVNetV1-39	49.0M	35.3	39.7	68
+ residual	49.0M	35.5 (+0.2)	39.8 (+0.1)	68
+ SE [13]	50.8M	34.6 (-0.7)	39.0 (-0.7)	70
+ eSE, *ours*	52.6M	35.6 (+0.3)	40.0 (+0.3)	70
VoVNetV1-57	63.0M	36.1	40.8	74
+ residual	63.0M	36.4 (+0.3)	41.1 (+0.3)	74
+ SE [13]	65.9M	35.9 (-0.2)	40.8	77
+ eSE, *ours*	68.9M	36.6 (+0.5)	41.5 (+0.7)	76
VoVNetV1-99	83.6M	31.5	35.3	101
+ residual	83.6M	37.6 (+6.1)	42.5 (+7.2)	101
+ SE [13]	88.0M	37.1 (+5.6)	41.9 (+6.6)	107
+ eSE, *ours*	96.9M	38.3 (+6.8)	43.5 (+8.2)	106

Table 4: **VoVNetV2** Start from VoVNetV1, VoVNetV2 is improved by adding residual connection [10] and the proposed effetive SE (eSE).

residual connection and the proposed effective SE (eSE) module into the VoVNet. Table 4 shows residual connection consistently improves VoVNet-39/57/99. In particular, the reason that the improved AP margin of VoVNet-99 is bigger than VoVNet-39/57 is that VoVNet-99 comprised of more OSA modules can have more effect of residual connection that alleviates the optimization problem.

To validate eSE, we also apply the SE [13] to the VoVNet and compare it with the proposed eSE. As shown in Table 4, the SE worsens the performance of VoVNet or has no effect because the diversified feature map of OSA module losses channel information due to channel dimension reduction in the SE. Contrary to the SE, our eSE maintaining channel information using only 1 FC layer boosts both AP_{mask} and AP_{box} from VoVNetV1 with slight computation.

Comparison to other backbones: We expand VoVNetV2 on various scales; large (V-99), base (V-39/57), and lightweight (V-19) which correspond to ResNeXt-32-8d [36] & HRNet-W48 [32], ResNet-50/101 [10] & HRNet-W18/W32 [32], and MobileNetV2 [31], respectively. Table 3 and Figure 1 (bottom) demonstrate VoVNetV2 is well-balanced backbone network in terms of accuracy and speed. While VoVNetV1-39 already outperforms its counterparts, VoVNetV2-39 shows better performance than ResNet-50/HRNet-W18 by a large margin of 1.2%/2.6% at faster speeds, respectively. Especially, the gain of AP^{box} is bigger than AP^{mask}, 1.5%/3.3%, respectively. A similar result pattern is shown in VoVNetV2-57 with its counterparts.

For large model, showing much faster run time (1.5×), VoVNetV2-99 achieves competitive AP^{mask} or higher AP^{box} than ResNeXt-101-32x8d despite fewer model parameters. For small model, VoVNetV2-19 outperforms MobileNetV2 by a large margin of 1.7% AP^{mask}/3.3%AP^{box}, with comparable speed.

3.2. Comparison with state-of-the-arts methods

For further validation of the CenterMask, we compare the proposed CenterMask with state-of-the-art instance segmentation methods. As most methods [25, 5, 8, 1, 7] use train augmentation, we also adopt the scale-jitter where the shorter image side is randomly sampled from [640, 800] pixels [8]. For Centermask-Lite, [580, 600] scale jittering is used for training. We train CenterMask and CenterMask-Lite for 24/36 epochs and 48 epochs, respectively. Note that we do not use test-time augmentation [8] (multi-scale). The other hyper-parameters are kept same as ablation study. For fair speed comparison, we inference models on the same GPU as counterparts. Specifically, since most large models are tested on V100 GPU and YOLACT [1] models are reported on Titan Xp GPU, we also report CenterMask models on V100 and CenterMask-Lite models on Xp.

Method	Backbone	epochs	AP^{mask}	AP_S^{mask}	AP_M^{mask}	AP_L^{mask}	AP^{box}	AP_S^{box}	AP_M^{box}	AP_L^{box}	Time	FPS	GPU
Mask R-CNN, *ours*	R-101-FPN	24	37.9	**18.1**	40.3	53.3	42.2	24.9	45.2	52.7	94	10.6	V100
ShapeMask [17]	R-101-FPN	N/A	37.4	16.1	40.1	53.8	42.0	24.3	45.2	53.1	125	8.0	V100
TensorMask [5]	R-101-FPN	72	37.1	17.4	39.1	51.6	-	-	-	-	380	2.6	V100
RetinaMask [7]	R-101-FPN	24	34.7	14.3	36.7	50.5	41.4	23.0	44.5	53.0	98	10.2	V100
CenterMask	R-101-FPN	24	**38.3**	17.7	**40.8**	**54.5**	**43.1**	**25.2**	**46.1**	**54.4**	**72**	**13.9**	V100
CenterMask*	R-101-FPN	36	39.8	21.7	42.5	52.0	44.0	25.8	46.8	54.9	66	15.2	V100
Mask R-CNN, *ours*	X-101-FPN	36	39.3	19.8	41.4	55.0	44.1	27.0	46.7	54.6	165	6.1	V100
CenterMask	X-101-FPN	36	39.6	19.7	42.0	55.2	44.6	27.1	47.2	55.2	123	8.1	V100
CenterMask	V-99-FPN	36	**40.6**	**20.1**	**42.8**	**57.0**	**45.8**	**27.8**	**48.3**	**57.6**	**84**	**11.9**	V100
CenterMask*	V-99-FPN	36	41.8	24.4	44.4	54.3	46.5	28.7	48.9	57.2	77	12.9	V100
YOLACT-400 [1]	R-101-FPN	48	24.9	5.0	25.3	**45.0**	28.4	10.7	28.9	**43.1**	22	45.5	Xp
CenterMask-Lite	M-v2-FPN	48	**26.7**	**9.0**	**27.0**	40.9	**30.2**	**14.2**	**31.9**	40.9	**20**	**50.0**	Xp
YOLACT-550 [1]	R-50-FPN	48	28.2	9.2	29.3	44.8	30.3	14.0	31.2	43.0	23	43.5	Xp
CenterMask-Lite	V-19-FPN	48	**32.4**	**13.6**	**33.8**	**47.2**	**35.9**	**19.6**	**38.0**	**45.9**	**23**	**43.5**	Xp
YOLACT-550 [1]	R-101-FPN	48	29.8	9.9	31.3	47.7	31.0	14.4	31.8	43.7	30	33.3	Xp
YOLACT-700 [1]	R-101-FPN	48	31.2	12.1	33.3	47.1	33.7	16.8	35.6	45.7	42	23.8	Xp
CenterMask-Lite	R-50-FPN	48	32.9	12.9	34.7	48.7	36.7	18.7	39.4	48.2	29	34.5	Xp
CenterMask-Lite	V-39-FPN	48	**36.3**	**15.6**	**38.1**	**53.1**	**40.7**	**22.4**	**43.2**	**53.5**	**28**	**35.7**	Xp

Table 5: **CenterMask** instance segmentation and detection performance on COCO test-dev2017. Mask R-CNN, RetinaMask, and CenterMask are implemented on the same base code [27] and CenterMask* is implemented on top of Detectron2[1] [35]. R, X, V, and M denote ResNet, ResNeXt-32x8d, VoVNetV2, and MobileNetV2, respectively. For fair comapriosn, these results are tested with *one thread* and single-scale.

Under the same ResNet-101 backbone, CenterMask outperforms all other counterparts in terms of both accuracy (AP^{mask}, AP^{box}) and speed. In particular, compared to RetinaMask [7] that has similar architecture (i.e., one-stage detector + mask branch), CenterMask achieves 3.6%AP^{mask} gain. In less than half training epochs, CenterMask also surpasses the dense sliding window method, TensorMask [5], by 1.2%AP^{mask} at ×5 faster speed. Furthermore, to the best of our knowledge, the CenterMask with VoVNetV2-99 is the first method to achieves 40% AP^{mask} at over 10 fps. It is noted that after first submission, Detectron2 [35] has been released that is a better baseline code. Thus, we also re-implement our CenterMask* on top of Detectron2 [35] and obtain further performance gain.

We also compare with YOLACT [1] that is the representative real-time instance segmentation. We use four kinds of backbones (e.g., MobileNetV2, VoVNetV2-19, VoVNetV2-39, and ResNet-50), which have a different accuracy-speed tradeoff. Table 5 and Figure 1 (top) demonstrate CenterMask-Lite is consistently superior to YOLACT in terms of accuracy and speed. Compared to YOLACT, all CenterMask-Lite models achieve over 30 fps speed with large margins of both AP^{mask} and AP^{box}.

4. Discussion

In Table 5, we observe that using the same ResNet-101 backbone, Mask R-CNN [9] shows better performance than CenterMask on small object. We conjecture that Mask R-CNN [9] uses larger feature maps (P2) than CenterMask (P3) in which the mask branch can extract much finer spatial layout of an object than the P3 feature map. We note that there are still rooms for improving one-stage instance segmentation performance like techniques [2, 3] of Mask R-CNN [9].

5. Conclusion

We have proposed a real-time anchor-free one-stage instance segmentation and more effective backbone networks. Adding spatial attention guided mask branch to the anchor-free one stage instance detection, CenterMask achieves state-of-the-art performance at real-time speed. The newly proposed VoVNetV2 backbone spanning from lightweight to larger models makes CenterMask well-balanced performance in terms of speed and accuracy. We hope CenterMask will serve as a baseline for real-time instance segmentation. We also believe our proposed VoVNetV2 can be used as a strong and efficient backbone network for various vision tasks [37, 16].

Acknowledgements. We thank Hyung-il Kim and the DeepView team in ETRI. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. B0101-15-0266, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis and No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network).

[1] After the initial submission, Detectron2 [35] has been released and we has developed the improved CenterMask* on top of the Detectron2 [35].

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 1, 2, 7, 8

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 1, 8

[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, pages 4974–4983, 2019. 1, 8

[4] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, pages 5659–5667, 2017. 3

[5] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollar. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019. 7, 8

[6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019. 2

[7] 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. 7, 8

[8] Kaiming He, Ross Girshick, and Piotr Dollar. Rethinking imagenet pre-training. In *ICCV*, 2019. 7

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 3, 5, 6, 8

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4, 7

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[12] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NIPS*, pages 9401–9411, 2018. 3

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 2, 3, 4, 7

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[15] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, pages 6409–6418, 2019. 1, 5, 6

[16] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *ICCV*, pages 1745–1753, 2019. 8

[17] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *ICCV*, 2019. 8

[18] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 2

[19] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *CVPR Workshops*, pages 0–0, 2019. 1, 2, 4, 7

[20] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *arXiv preprint arXiv:1901.01892*, 2019. 1

[21] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3, 5, 6

[22] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *ICCV*, 2017. 1, 3

[23] 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 *ECCV*, 2014. 1, 2, 5

[24] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. 1

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 7

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 3

[27] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark`, 2018. 8

[28] Zheng Qin, Zeming Li, Zhaoning Zhang, Yiping Bao, Gang Yu, Yuxing Peng, and Jian Sun. Thundernet: Towards real-time generic object detection on mobile devices. In *ICCV*, 2019. 3

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

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

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 7

[32] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019. 2, 7

[33] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 2, 3, 5, 6

[34] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018. 3

[35] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 8

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 2, 7

[37] Kimin Yun, Yongjin Kwon, Sungchan Oh, Jinyoung Moon, and Jongyoul Park. Vision-based garbage dumping action detection for real-world surveillance platform. *ETRI Journal*, 41(4):494–505, 2019. 8

[38] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 2

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

[40] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *ICCV*, 2019. 3