

Instance Segmentation and Object Detection with Bounding Shape Masks

Ha Young Kim^{1,2,*}, Ba Rom Kang²

¹ Department of Financial Engineering, Ajou University
Worldcupro 206, Yeongtong-gu, Suwon, 16499, Republic of Korea

² Department of Data Science, Ajou University
Worldcupro 206, Yeongtong-gu, Suwon, 16499, Republic of Korea

Abstract

Many recent object detection algorithms use the bounding box regressor to predict the position coordinates of an object (i.e., to predict four continuous variables of an object's bounding box information). To improve object detection accuracy, we propose four types of object boundary segmentation masks that provide position information in a different manner than that done by object detection algorithms. Additionally, we investigated the effect of the proposed object bounding shape masks on instance segmentation. To evaluate the proposed masks, our method adds a proposed bounding shape (or box) mask to extend the Faster R-CNN framework; we call this Bounding Shape (or Box) Mask R-CNN. We experimentally verified its performance with two benchmark datasets, MS COCO and Cityscapes. The results indicate that our proposed models generally outperform Faster R-CNN and Mask R-CNN.

Introduction

Object detection and instance segmentation are crucial tasks in computer vision. Object detection determines the class of each object instance and the location of each object instance in an image. Deep learning-based object detection approaches have recently achieved notable success, such as the Overfeat (Sermanet et al. 2013), Fast R-CNN (Girshick 2015), and Faster R-CNN (Ren et al. 2015) models. In addition, convolutional neural network (CNN) algorithms such as instance segmentation, combining object detection with semantic segmentation, used to predict to which class in the image each pixel (px) belongs to, have been actively studied (He et al. 2017; Dai, He, and Sun 2016; Dai et al. 2016). However, to apply these object recognition algorithms to real-world applications, many problems such as improving the performance and reducing the computational cost still need to be addressed.

Many recent object detection algorithms, including the object detection representative method Faster R-CNN, comprise a classifier for the object class and a bounding box regressor for predicting the locations of object instances. Bounding box regressors are not easy to predict from the four-dimensional continuous variables (x-coordinate, y-coordinate, width, and height) in images. Teaching the network with more information to predict a bounding box will

improve object detection performance. Mask R-CNN (He et al. 2017) improves the accuracy of bounding box prediction by adding object mask information to Faster R-CNN. Therefore, the performance of object detection will be improved by also teaching the boundary information of the object in the form of a mask. We define bounding shape masks and bounding box masks as shown in Figure 1.

We propose a method to improve bounding box prediction accuracy by adding a bounding shape mask branch to the Faster R-CNN framework. This method will be called *bounding shape mask R-CNN (BshapeMask)*. We also study the effect of instance segmentation on bounding shape masks using only object boundary information, unlike the instance segmentation branch of Mask R-CNN. Figure 2 shows the proposed framework. Similarly, we call the framework that adds a bounding box mask to the Faster R-CNN framework the *bounding box mask R-CNN (BboxMask)*.

In this study, we used four masks, *Thick bounding shape mask* (Figure 1 (b)), *Scored bounding shape mask* (Figure 1 (c)), *Thick bounding box mask* (Figure 1 (f)), and *Scored bounding box mask* (Figure 1 (g)) instead of Figure 1 (a) and (e) and the models using each mask are called *Thick BshapeMask*, *Scored BshapeMask*, *Thick BboxMask*, and *Scored BboxMask*, respectively. Also, as shown in Figure 1 (d) and (h), various thickness masks were used to evaluate the accuracy according to the object boundary thickness. More details are given in Section *Bounding Shape Mask R-CNN*.

As in the proposed framework, the model not only learns the coordinate information of the bounding box through the bounding box regressor branch, but also learns the bounding box information expressed in a different way, which is the boundary information of object instances. Therefore, the conjecture of this study is that BshapeMask will improve bounding box prediction accuracy because it provides various information about the bounding box.

The Thick bounding shape (or box) mask branch is used to perform pixel-wise binary classification to detect object boundaries because the thick boundary of the object is filled with 1s, while objects inside and the background are filled with 0s. On the other hand, the Scored bounding shape (or box) mask branch performs pixel-wise regression because the thick boundary of the object is filled with various continuous values. Furthermore, we think it is more effective to use a Scored bounding shape mask than a Thick bounding shape

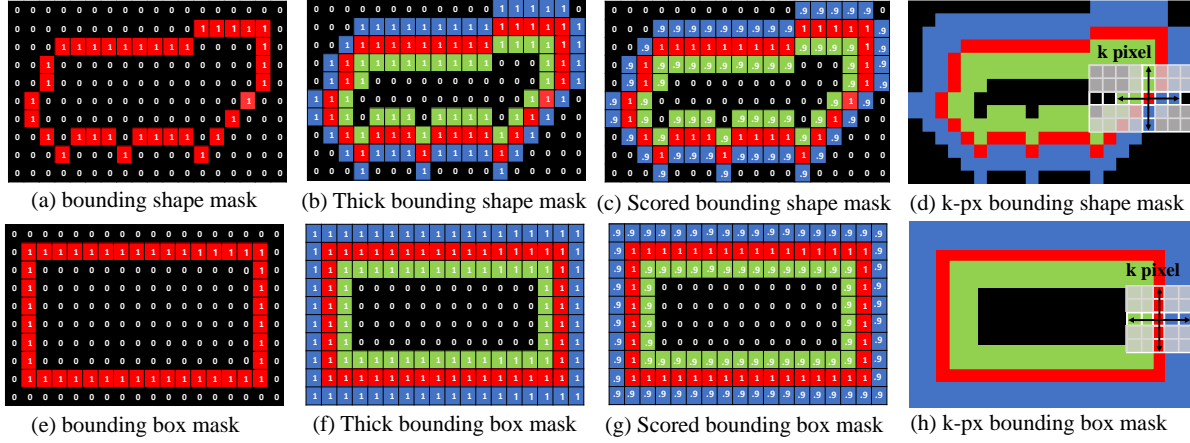


Figure 1: Our proposed bounding shape masks and bounding box masks. We used four thick boundary masks, which are those shown in (b), (c), (f), and (g) instead of (a) and (e) as a training issue. As in (d) and (h), varied-thickness masks were used to evaluate the accuracy achieved based on the object boundary thickness.

mask that uses the same value without distinguishing true boundaries from false boundaries by thickening the boundary. In addition, because edge or contour information is important to distinguish objects from background, we think that using a bounding shape mask rather than a bounding box mask will help to improve bounding box prediction accuracy.

We also believe that the proposed model will help improve instance segmentation performance. Because the boundary of an object is the criterion for dividing foreground and background, we assumed that the pixels near the boundary of the object are important information in distinguishing objects. We also think that the proposed model can learn more effectively than if learning all objects by focusing on learning the boundaries of objects.

To evaluate the proposed approaches, MS COCO (Lin et al. 2014) and the Cityscapes (Cordts et al. 2016) datasets were used. The BboxMask and BshapeMask results are presented to compare the effects of applying a bounding box mask and a bounding shape mask. We extended the boundaries of the object shape for training and experimented to make the true boundary thicker by various pixel amounts (3–11 pixels) to evaluate how thick it should be. Finally, we compared thick BshapeMask (BboxMask) and scored BshapeMask (BboxMask) to analyze the effect of a scored bounding shape (or box) mask. As a result, Scored BshapeMask had the highest performance for both benchmark datasets. This result also surpasses the performances of Faster R-CNN and Mask R-CNN in (He et al. 2017).

Related Work

Object Detection

In recent years, many studies have shown excellent performance in object detection (Liu et al. 2016; Ren et al. 2015; Redmon et al. 2016). Remarkable development of object detection began with Overfeat and region-based convolution neural network (R-CNN) (Girshick et al. 2014) ap-

proaches which are deep CNN-based models. In particular, the R-CNN methodology proposed in 2014 remains outstanding. Unlike Overfeat (e.g., sliding-window detection algorithms), R-CNN selects proposals including objects and classifies objects using the selected proposals. To solve the heavy calculations of R-CNN, Fast R-CNN and Faster R-CNN were developed. Fast R-CNN uses the *selective search* (Uijlings et al. 2013) using CNN’s feature map. Fast R-CNN uses a Region of Interest (RoI) pooling layer (Girshick 2015) to eliminate repeatedly feeding RoIs back into the CNN. Faster R-CNN uses Region Proposal Networks (RPNs) (Ren et al. 2015) to detect RoIs in the structure of Fast R-CNN. YOLO (Redmon et al. 2016) and SSD (Liu et al. 2016) have no additional network to detect RoIs and perform object proposal and object detection simultaneously and have advantages of reduced calculation and fast inference time.

Instance segmentation

Instance segmentation is instance-wise semantic segmentation and a segmentation approach to classify a single mask based on a proposal object. DeepMask (Pinheiro, Collobert, and Dollár 2015) predicts where an object exists by segmenting an object to propose candidates. Dai et al. (2016) proposed a cascading stage model using a shared feature map to segment proposed instance(s) from a bounding box. Fully convolutional instance-aware semantic segmentation (FCIS) (Li et al. 2016) used the relative position of a feature through the position-sensitive score map using an improvement (Dai et al. 2016) for the instance segmentation. FCIS simultaneously predicts the bounding box and mask, but FCIS has a systemic error with overlapping objects and only roughly detects the bounding of objects. Mask R-CNN uses an elaborate instance segmentation result. This model uses RoIAlign (He et al. 2017) to obtain precise RoIs and uses them together with three branches (classification, bounding box detection, and segmentation) to achieve good performance. These methods focus on the inside of the proposal

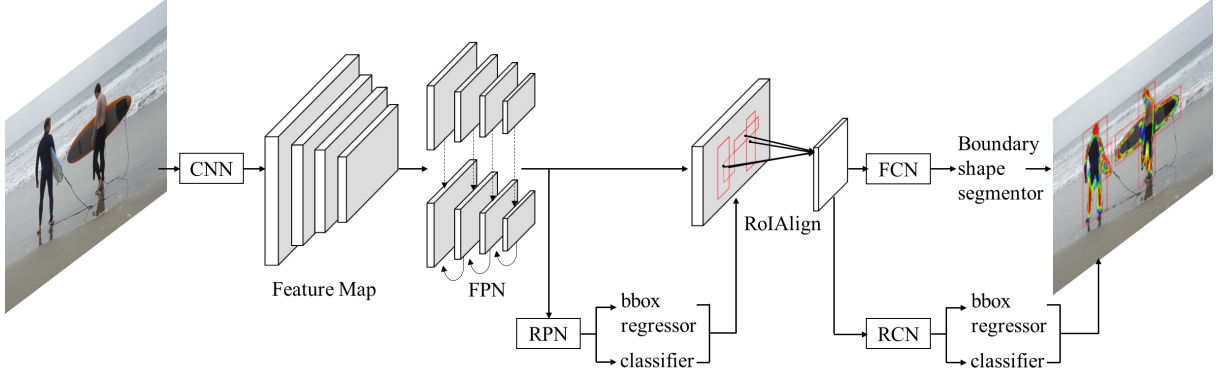


Figure 2: Bounding Mask R-CNN Framework, named BshapeMask. We extend Faster R-CNN by adding a bounding shape mask branch. The BshapeMask consists of RPN (a classifier and a bounding box regressor), RCN (a classifier and a bounding box regressor), and FCN (a bounding shape segmentor).

object for instance segmentation. In contrast, we study the edges of objects (information that distinguish between foreground and background).

Bounding Shape Mask R-CNN

Bounding Shape (or Box) Mask Representation

As mentioned in the Introduction, we define bounding shape masks and bounding box masks to make the position information of the bounding box of an object different from the coordinate information, as shown in Figure 1. In the first row of Figure 1, we define the object’s boundary with various masks, and we define the object’s bounding box with other masks in the second row. Except for the object boundary and bounding box information, we define the masks in the same way, so we explain them using the bounding shape mask example. As shown in Figure 1 (a), the raw boundary shape mask consists of only a one-pixel width true object boundary. But in this case, the amount of information is too low and learning is difficult. Thus, we use a thicker boundary using several pixels, as in Figure 1 (b) and (c); we defined the thicker boundary by stretching the boundaries inside and outside of the true boundary. As seen in Figure 1 (d) and (h), both the k -px Thick bounding shape (or box) mask and the k -px Scored bounding shape (or box) mask are the extend both the inside and outside of the true boundary by k pixels, so the thickness of the boundary is $2k + 1$. However, the way the boundary is expanded is different. We call expanded boundaries *false boundary pixels* because they are not real boundary pixels.

In the Thick bounding shape mask, false boundary pixels are filled with 1s, which is the same as for true boundary pixels. The mathematical expression of this mask is as follows. Let $M = (m_{ij}) \in \mathbb{R}^{v \times w}$ be a true boundary mask matrix, $B = \{(i, j) : m_{ij} = 1, 1 \leq i \leq v, 1 \leq j \leq w\}$ is a true boundary index set, and $X = (x_{pq}) \in \mathbb{R}^{v \times w}$ is a k -px Thick bounding shape (or box) mask matrix. For $\forall(i, j) \in B$,

$$\begin{cases} x_{pj} = 1, & \text{if } 1 \leq i - k \leq p \leq i + k \leq v \\ x_{iq} = 1, & \text{if } 1 \leq j - k \leq q \leq j + k \leq w. \end{cases} \quad (1)$$

All remaining values are filled with zeros. In this case, it is difficult to distinguish between the false boundary and true boundary, so we propose the Scored bounding shape mask because we can learn boundaries more effectively by defining false boundary and true boundary values differently.

The value of the false boundary is reduced at a constant rate in proportion to the distance from the actual boundary. Thus, for the Scored bounding shape mask, false boundary pixels are filled with distance-based scored numbers, which are positive numbers less than 1, to generate a mask. Let $Y = (y_{pq}) \in \mathbb{R}^{v \times w}$ be a k -px Scored bounding shape (or box) mask matrix. We use the same matrices M and B as Eq.(1), and s is a predetermined positive constant (less than 1) that controls the magnitude by which the value decreases. Then for $\forall(i, j) \in B$,

$$\begin{cases} y_{pj} = 1 - d_1 s, & \text{if } 1 \leq i - k \leq p \leq i + k \leq v \\ y_{iq} = 1 - d_2 s, & \text{if } 1 \leq j - k \leq q \leq j + k \leq w, \end{cases} \quad (2)$$

where $d_1 = |p - i|$ and $d_2 = |q - j|$ are the distance from the true boundary, and the remaining values are filled with zeros. We set s as 0.05.

Framework

The BshapeMask is a framework which combines Faster R-CNN with a bounding shape mask branch. Specifically, we use a bounding shape mask branch instead of the instance segmentation mask branch in Mask R-CNN. As shown in Figure 2, our proposed network consists of a convolutional backbone network (a feature extractor), the RPN, RCN, and the bounding box branch based on FCN (?). In the backbone, we use the Feature Pyramid Network (FPN) (Lin et al. 2017) to improve detection accuracy. The BshapeMask flow is as follows. First, the CNN extracts feature maps from an image. We use ResNet50 or ResNet101 as this backbone. FPN combines feature maps in each layer of the CNN except the first layer and forwards combined feature maps to the RPN. The RPNs two branches (classifier and bounding box regressor), propose the RoIs. Objects are detected using the RoIs through the RCN. The boundary (or box) mask branch, which is also the object boundary segmentor, uses

RoIs at the same time as the RCN. Through this process, object boundaries are segmented, and the architecture of the FCN used in this branch has the same architecture as the Mask R-CNN. BboxMask uses the same framework except that it uses a bounding box mask branch.

Training BshapeMask/BboxMask

The loss function is the same in both BshapeMask and BboxMask. Therefore, we describe it with BshapeMask. However, the Thick model and the Scored model have different losses. The Thick BshapeMask is trained using three losses, which are L_{RPN} , L_{RCN} , and L_{Tmask} , from the three branches (RPN, RCN and Thick bounding shape mask branch, respectively). Scored BshapeMask uses L_{Smask} , the Scored bounding shape mask loss, instead of L_{Tmask} . Our total loss function for training our proposed model, Thick BshapeMask, is defined as follows:

$$Loss_{total} = \alpha L_{RPN} + \beta L_{RCN} + \gamma L_{Tmask}, \quad (3)$$

where α , β , and γ are predetermined positive constants and losses of RPN and RCN and are the same as those for Faster R-CNN. The total loss of Scored BshapeMask uses L_{Smask} instead of L_{Tmask} .

Because the ground-truth only consists of 0 and 1 in the Thick bounding shape (or box) mask, it is a pixel-wise binary classification problem. Thus, we use the binary cross-entropy loss and Thick bounding shape mask loss for each RoI as follows: $L_{Tmask} = -\frac{1}{HW} \sum_i^W \sum_j^H \{t_{ij} \log \hat{t}_{ij} + (1 - t_{ij}) \log(1 - \hat{t}_{ij})\}$, where t_{ij} is the ground-truth label of the mask and the predicted value is \hat{t}_{ij} , and H and W indicate the height and width of the mask, respectively.

The ground-truth values of the Scored bounding shape mask have continuous values from 0 to 1. Thus, Scored BshapeMask and Scored BboxMask solve the pixel-wise regression problem while Thick BshapeMask and Thick BboxMask solve a pixel-wise classification problem. Thus, we use the Euclidean loss with the same notation as L_{Tmask} as follows: $L_{Smask} = \frac{1}{2HW} \sum_i^W \sum_j^H (t_{ij} - \hat{t}_{ij})^2$.

Post-processing

Post-processing is required for instance segmentation, because our proposed bounding shape masks only segment boundaries of objects. We performed post-processing of the two-step procedures. The first step is to connect the predicted boundary and fill it. To connect and fill the interior of the predicted mask, we used a modified Prims algorithm (Prim 1957), which is one of the techniques used in the Minimum Spanning Tree. The second step is the process of reducing the thickness of the outer and inner boundaries. For this, we use the fully connected CRFs-based mean field approximation (Krähenbühl and Koltun 2011).

Experiment

We experimented to investigate the following three topics. The first was to identify which of the proposed masks is most effective. The second was to study if the bounding box prediction performance improves by adding the proposed mask to Faster R-CNN. Finally, we evaluated whether

the proposed model improves the performance of instance segmentation when the model learns only the boundary of the object, not the entire object. To verify our proposed methods, we used two benchmark datasets, MS COCO and Cityscapes. For each dataset, we compared bounding box prediction and instance segmentation prediction performance for the four proposed models (Thick BshapeMask, Scored BshapeMask, Thick BboxMask, and Scored BboxMask) using various boundary thicknesses. In addition, to evaluate the performance of the bounding shape (or box) masks, we compared the results of the proposed model with Faster R-CNN and Mask R-CNN.

For fair comparison of our proposed models with Faster R-CNN and Mask R-CNN, we used the same hyperparameters as Faster R-CNN and Mask R-CNN except those related to the bounding shape (or box) mask branch except learning rate and minibatch size owing to our limited experimental environment (2 NVIDIA GTX 1080Ti GPUs). In the Mask R-CNN paper, they trained a model with eight NVIDIA Tesla M40 GPUs. To train our models, each image contained 200 sampled RoIs, with a ratio of 1:3 of foreground to background, and anchors of five different scales and three aspect ratios are used as in (Ren et al. 2015). All experiments in this study were performed with ResNet50 or ResNet101 as the backbone feature extractor models.

We evaluate our models using the mean average precision (mAP) with 0.5 intersection-over-union (IoU). We follow the bounding box evaluation using the Pascal VOC evaluation metric and the instance segmentation evaluation using the MS COCO evaluation metric.

The MS COCO Dataset

We resized images such that their shorter edge is 800 pixels. During training, we set the initial learning rate to 0.001 and divided it by 10 when the loss curve was convergent (down to 0.00001). The minibatch size was 4. For the backbone (ResNet50 or ResNet101), we used pre-trained ImageNet (?) weights. The experimental results are shown in Table 1 and Table 2. Table 1 shows the bounding box prediction accuracy of the proposed models with various thicknesses and Table 2 shows the instance segmentation accuracy of the proposed models. Figure 3 shows examples of bounding box segmentation results for a 7-px Scored BboxMask (ResNet101) and Figure 4 shows examples of instance segmentation and object detection results of a 7-px Scored BshapeMask (ResNet101). Pixels in the predicted box are blue for values of 0.5 or more and less than 0.6, green for values of 0.6 or more and less than 0.7, yellow for values of 0.7 or more and less than 0.8, and red for values of 0.8 or more. The same color scheme is used in Figure 5. The experimental results with four thicknesses (three, five, seven and eleven pixels) of object boundaries, show that the best bounding box prediction accuracy and instance segmentation accuracy was achieved with seven pixels among all models.

Comparing BboxMask and BshapeMask results, BshapeMask is generally more accurate, as we expected. The boundary shape information of the object helps in object detection much more effectively than the bounding box infor-



Figure 3: Examples of bounding box segmentation results of 7-px Scored BboxMask (ResNet101) on the MS COCO val dataset. It can be seen that objects’ bounding boxes are well segmented even though no post-processing has been performed.

Table 1: Bounding box prediction results on MS COCO validation dataset with mAP (%). Star (*) means the validation result in Mask R-CNN paper (He et al. 2017). The bounding box detection accuracy is denoted by mAP_{bb} .

Model(ResNet50)	mAP_{bb}	Model(ResNet50)	mAP_{bb}
3-px Thick BboxMask	57.7	3-px Scored BboxMask	57.9
5-px Thick BboxMask	58.2	5-px Scored BboxMask	58.4
7-px Thick BboxMask	58.9	7-px Scored BboxMask	58.7
11-px Thick BboxMask	57.5	11-px Scored BboxMask	57.5
3-px Thick BshapeMask	58.1	3-px Scored BshapeMask	59.2
5-px Thick BshapeMask	59.3	5-px Scored BshapeMask	62.4
7-px Thick BshapeMask	59.5	7-px Scored BshapeMask	62.6
11-px Thick BshapeMask	59.1	11-px Scored BshapeMask	61.1
Faster R-CNN	57.4		
Mask R-CNN	58.0		
Model(ResNet101)	mAP_{bb}	Model(ResNet101)	mAP_{bb}
3-px Thick BboxMask	60.1	3-px Scored BboxMask	64.8
5-px Thick BboxMask	60.8	5-px Scored BboxMask	68.5
7-px Thick BboxMask	60.5	7-px Scored BboxMask	69.1
11-px Thick BboxMask	59.9	11-px Scored BboxMask	68.9
3-px Thick BshapeMask	61.1	3-px Scored BshapeMask	65.3
5-px Thick BshapeMask	61.7	5-px Scored BshapeMask	68.3
7-px Thick BshapeMask	62.4	7-px Scored BshapeMask	69.9
11-px Thick BshapeMask	61.5	11-px Scored BshapeMask	69.7
Faster R-CNN*	59.6		
Mask R-CNN*	60.3		

mation. The results reveal that Scored models generally perform better than Thick models. As mentioned earlier, this experimentally confirms that filling the boundaries differently from the true boundary values improves the object detection performance, rather than filling them all with the same value when thickening the boundary.

In further detail, among the models using ResNet50 as a backbone, the 7-px Scored BshapeMask (ResNet50) with a bounding box mAP of 62.6% and a mask mAP of 56.6% achieved the best performance, as shown in Tables 1 and 2. Similarly, among the models using ResNet101 as a backbone, the 7-px Scored BshapeMask (ResNet101) showed the best bounding box prediction performance with 69.9% (mAP) and instance segmentation performance with 58.7%

Table 2: Instance segmentation results on MS COCO validation dataset with mAP (%). Star (*) means the validation result in the Mask R-CNN paper (He et al. 2017). Instance segmentation accuracy mAP is denoted by mAP_{mask} .

Model(ResNet50)	mAP_{mask}	Model(ResNet50)	mAP_{mask}
3-px Thick BshapeMask	48.1	3-px Scored BshapeMask	51.1
5-px Thick BshapeMask	51.1	5-px Scored BshapeMask	55.3
7-px Thick BshapeMask	50.2	7-px Scored BshapeMask	56.6
11-px Thick BshapeMask	50.1	11-px Scored BshapeMask	55.9
Mask R-CNN	55.2		
Model(ResNet101)	mAP_{mask}	Model(ResNet101)	mAP_{mask}
3-px Thick BshapeMask	49.8	3-px Scored BshapeMask	51.2
5-px Thick BshapeMask	53.4	5-px Scored BshapeMask	57.5
7-px Thick BshapeMask	52.1	7-px Scored BshapeMask	58.7
11-px Thick BshapeMask	50.8	11-px Scored BshapeMask	57.9
Mask R-CNN*	57.3		

(mAP).

We also compared the results of our models with those of Faster R-CNN and Mask R-CNN. Prior to the comparison, the accuracies for Faster R-CNN (ResNet50) and Mask R-CNN (Res-Net50) were not reported in (He et al. 2017; Ren et al. 2015), so the accuracies shown in Table 1 are the results of our experiments, while Faster R-CNN (ResNet101) and Mask R-CNN (Res-Net101) are the accuracies reported in (He et al. 2017).

First, when comparing models using ResNet50 as a backbone, all our models outperform Faster R-CNN. Further, compared to the Mask R-CNN with a bounding box accuracy of 58.0% (mAP), almost of all of our models have better performance. Three of our models (5, 7, 11-px Scored BshapeMask models) outperform Mask R-CNN with 55.2% (mAP) instance segmentation accuracy. Especially, the 7-px Scored BshapeMask (ResNet50) with a bounding box accuracy of 62.6% (mAP) is 2.3 points higher than the Mask R-CNN (Res-Net101) using about twice as deep a backbone.

Second, when comparing models using ResNet101 as a backbone, all our models outperform the 59.6% (mAP) bounding box accuracy of Faster R-CNN, as shown in Table 1. Further, compared to Mask R-CNN (ResNet101) with a bounding box accuracy of 60.3% (mAP), except for two models (3 and 11-px Thick BboxMask), our models

achieved better performance. Similar to the results of previous models using ResNet50, three of our models (5, 7, and 11-px Scored BshapeMask) outperform the 57.3% (mAP) instance segmentation accuracy of Mask R-CNN. The 7-px Scored BshapeMask (ResNet101) is best model among our models and its bounding box accuracy is 9.6 points higher than the Mask R-CNN (ResNet101), which is an improvement of about 15.9%. And the instance segmentation accuracy of the 7-px Scored BshapeMask is 1.4 points higher than the Mask R-CNN (ResNet101).

The Cityscapes Dataset

The Cityscapes *trainvaltest* dataset is composed of a 2,975 image training set and 500 image validation set. The cityscape dataset has nine classes including background: person, rider, car, truck, bus, train, motorcycle, and bicycle. During training, we set the initial learning rate to 0.001 and then divided by 10 when the loss curve was convergent (down to 0.00001); we used a minibatch size of 2 and the pre-trained MS COCO weights for backbones. Although the raw data size was 2048×1024 pixels, it was reduced to 1024×1024 pixels for operating in an 11 GB RAM environment. We performed the same experiments and evaluated the bounding box accuracy and instance segmentation accuracy in the same manner as the MS COCO dataset. The results are summarized in Table 3 and 4.

For bounding box prediction, the BboxMask models were also less accurate than the BshapeMask models and the Scored models were more accurate than Thick models (same as for COCO). The 3-px Scored BshapeMask (ResNet50), the most accurate bounding box model among our models, achieved a bounding box accuracy of 55.3%, which is 10.1 points higher than the Faster R-CNN (ResNet50), which has an accuracy of 45.2% (mAP). This result is also 15.2% better than Mask R-CNNs accuracy of 48.0%. In addition, all our models achieved higher object detection accuracy than Faster R-CNN (ResNet50), and except for one model, the BshapeMask (ResNet50) models outperformed Mask R-CNN (ResNet50). Comparing object detection results of our models using ResNet101 with Faster R-CNN (ResNet101) and Mask R-CNN (ResNet101), all our models achieved higher detection accuracy. The highest performance model, 3-px Scored BshapeMask (ResNet101), achieved a bounding box prediction accuracy of 58.9% (mAP), which is 30% better than Faster R-CNN (ResNet101) and 21.2% better than Mask R-CNN (ResNet101).

Regarding instance segmentation accuracy, all proposed models, except for the 3-px and 7-px Thick BshapeMask, show higher instance segmentation accuracy among the proposed models using ResNet50 as a backbone. Also, except for 7-px Thick BshapeMask, all proposal models that used ResNet101 as the backbone show higher instance segmentation accuracy than Mask R-CNN. Especially, 11-px Thick BshapeMask with ResNet101 has an instance segmentation accuracy of 52.4% (mAP) which is 1.7 points higher than Mask R-CNN and is an improvement of 3.3%. The 3-px Scored BshapeMask with ResNet101 has instance segmentation accuracy of 54.2% (mAP) which is 3.5 points higher than Mask R-CNN and is an improvement of 6.9%. In ad-

Table 3: Bounding box prediction results on Cityscapes validation dataset with mAP (%). The bounding box prediction accuracy is denoted by mAP_{bb} .

Model(ResNet50)	mAP_{bb}	Model(ResNet50)	mAP_{bb}
3-px Thick BboxMask	45.5	3-px Scored BboxMask	47.6
5-px Thick BboxMask	45.8	5-px Scored BboxMask	47.5
7-px Thick BboxMask	46.3	7-px Scored BboxMask	47.7
11-px Thick BboxMask	45.5	11-px Scored BboxMask	45.5
3-px Thick BshapeMask	47.7	3-px Scored BshapeMask	55.3
5-px Thick BshapeMask	48.2	5-px Scored BshapeMask	55.2
7-px Thick BshapeMask	48.9	7-px Scored BshapeMask	51.7
11-px Thick BshapeMask	49.9	11-px Scored BshapeMask	55.2
Faster R-CNN	45.2		
Mask R-CNN	48.0		
Model(ResNet101)	mAP_{bb}	Model(ResNet101)	mAP_{bb}
3-px Thick BboxMask	47.1	3-px Scored BboxMask	49.9
5-px Thick BboxMask	47.8	5-px Scored BboxMask	48.7
7-px Thick BboxMask	48.9	7-px Scored BboxMask	49.4
11-px Thick BboxMask	46.9	11-px Scored BboxMask	46.2
3-px Thick BshapeMask	48.8	3-px Scored BshapeMask	58.9
5-px Thick BshapeMask	49.3	5-px Scored BshapeMask	57.2
7-px Thick BshapeMask	50.1	7-px Scored BshapeMask	54.9
11-px Thick BshapeMask	51.5	11-px Scored BshapeMask	56.7
Faster R-CNN	45.3		
Mask R-CNN*	48.6		

Table 4: Instance segmentation results on Cityscapes validation dataset with mAP (%). Instance segmentation accuracy mAP is denoted by mAP_{mask} .

Model(ResNet50)	mAP_{mask}	Model(ResNet50)	mAP_{mask}
3-px Thick BshapeMask	42.3	3-px Scored BshapeMask	46.9
5-px Thick BshapeMask	46.6	5-px Scored BshapeMask	46.5
7-px Thick BshapeMask	45.1	7-px Scored BshapeMask	46.7
11-px Thick BshapeMask	46.7	11-px Scored BshapeMask	46.8
Mask R-CNN	46.4		
Model(ResNet101)	mAP_{mask}	Model(ResNet101)	mAP_{mask}
3-px Thick BshapeMask	51.1	3-px Scored BshapeMask	54.2
5-px Thick BshapeMask	51.4	5-px Scored BshapeMask	52.7
7-px Thick BshapeMask	49.1	7-px Scored BshapeMask	52.2
11-px Thick BshapeMask	52.4	11-px Scored BshapeMask	51.9
Mask R-CNN	50.7		

dition to the numerical results, we can compare the performance of our proposed models in Figure 5, where the first row is the result of a 3-px Scored BshapeMask (ResNet101) in the Cityscapes dataset and the second row is the result of Mask R-CNN (ResNet101). The proposed model not only can detect distant objects or overlapping objects more accurately than Mask R-CNN, but also grasps the edges of objects well.



Figure 4: Examples of object detection and instance segmentation results (without post-processing) for the 7-px Scored BshapeMask (ResNet101) on the MS COCO val dataset. Scored BshapeMask detects the outlines of objects and the bounding box well.

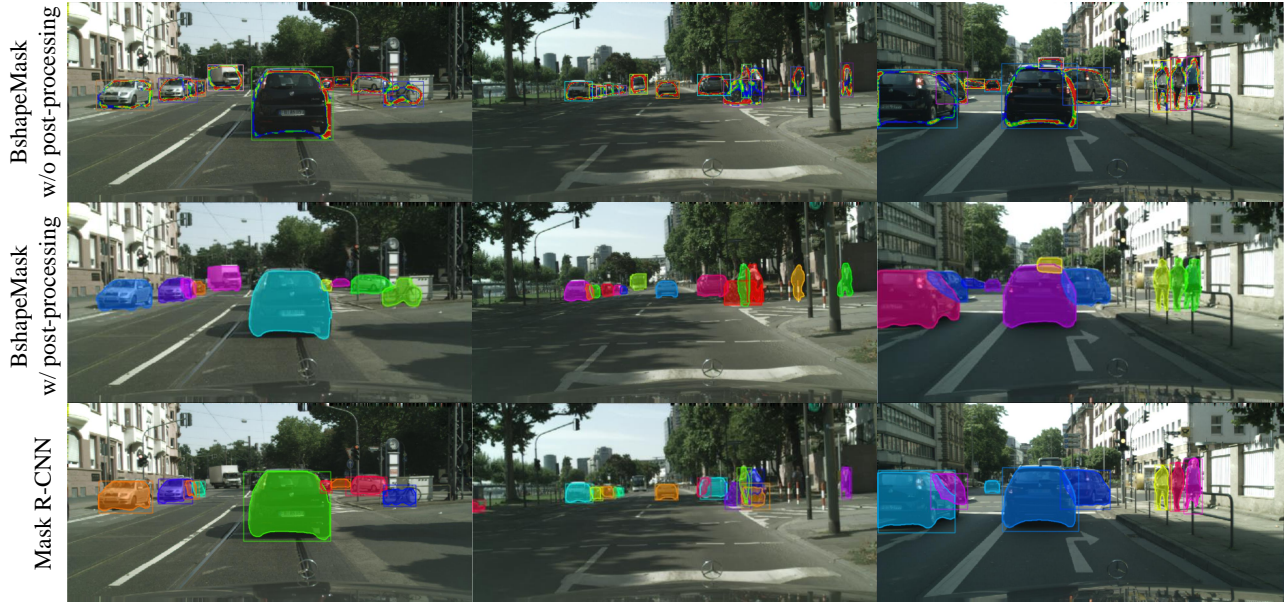


Figure 5: Comparison of the results of object detection and instance segmentation of CityScapes images by the proposed model (3-px Scored BshapeMask without post-processing (top) and 3-px Scored BshapeMask with post-processing (middle)) and the Mask R-CNN (bottom). Both models are trained in the same experimental environment and under the ResNet101 backbone. The proposed model detects objects that are not detected by Mask R-CNN as well as occluded objects.

Conclusion

In this paper, we define new masks that describe the boundary of object(s). To verify how these proposal masks can help in bounding box prediction and instance segmentation, we proposed new frameworks. We obtained the following three results. First, the Scored BshapeMask achieves the best performance among all frameworks. The second is that our proposed models displayed higher bounding box prediction accuracy than Faster R-CNN in all experiments, and confirmed that the proposed masks help to improve object de-

tection performance. Finally, almost all Scored BshapeMask models were better than Mask R-CNN, and this showed that intensive learning of object boundaries is useful for instance segmentation.

Acknowledgments

This research was supported by System LSI Business, Samsung Electronics Co., Ltd.

References

- Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.
- Dai, J.; He, K.; Li, Y.; Ren, S.; and Sun, J. 2016. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, 534–549. Springer.
- Dai, J.; He, K.; and Sun, J. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3150–3158.
- Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.
- Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2980–2988. IEEE.
- Krähenbühl, P., and Koltun, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, 109–117.
- Li, Y.; Qi, H.; Dai, J.; Ji, X.; and Wei, Y. 2016. Fully convolutional instance-aware semantic segmentation. *arXiv preprint arXiv:1611.07709*.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.
- Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*, volume 1, 4.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37. Springer.
- Pinheiro, P. O.; Collobert, R.; and Dollár, P. 2015. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, 1990–1998.
- Prim, R. C. 1957. Shortest connection networks and some generalizations. *Bell system technical journal* 36(6):1389–1401.
- Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.
- Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.
- Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; and LeCun, Y. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.
- Uijlings, J. R.; Van De Sande, K. E.; Gevers, T.; and Smeulders, A. W. 2013. Selective search for object recognition. *International journal of computer vision* 104(2):154–171.