This ACCV 2020 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2020
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

Point Proposal based Instance Segmentation with Rectangular Masks for Robot Picking Task

Satoshi Ito and Susumu Kubota

Corporate Research and Development Center, Toshiba Corporation, Japan
{satoshi13.ito,susumu.kubota}@toshiba.co.jp

Abstract. In this paper, we focus on instance segmentation of a top-view image for robot picking task. One difficulty in this setting is that objects are located in various orientations and highly overlapped, where a traditional box proposal approach such as Mask R-CNN does not work well because more than one objects often have very similar bounding-boxes. To address this issue, we adopt a recently developed point proposal approach. This approach firstly generates point proposals instead of box proposals, then an instance mask is predicted over an image for each proposal point. This procedure enables us to obtain pixel-precise masks even for objects sharing the same bounding-box. However, mask prediction over an image may produce a few false positive pixels apart from objects and these false positives are problematic for robot picking task. To suppress them, we introduce rectangular masks. A rectangular mask for each proposal point restricts the existence area of the corresponding object within the rectangle. The experimental result on WISDOM dataset shows that our method achieves superior performance to Mask R-CNN with the same backbone model and introduction of rectangular masks gives small improvement of mask AP and large improvement of box AP.

1 Introduction

With the recent increase of e-commerce, automation technologies using robots to reduce logistics costs attract great attention. In order to grasp and move objects, it's important for robots to understand each object's shape and location accurately. The task of segmenting objects in an image is called instance segmentation in computer vision. A lot of instance segmentation methods have been proposed so far and the detection performance has been much improved with recent advances of deep learning techniques. However, most of the methods were evaluated on benchmark datasets consisting of natural scenes such as Microsoft COCO [1] and Cityscapes [2] while instance segmentation of a top-view image is usually required for robot picking task. Therefore, those methods are not necessarily suitable for this task.

In this work, we study instance segmentation of a top-view image for robot picking task. Segmenting objects in a top-view image is difficult because objects can be located in various orientations and highly overlapped. In this situation,

since object shape cannot be approximated with an axis-aligned bounding-box, a traditional box proposal approach such as Mask R-CNN [3] does not work well. Furthermore, overlapped objects often have very similar bounding-boxes or even share the same bounding-box, and a box proposal approach cannot distinguish them.

To overcome the above problem, we use recently developed point proposal based methods [4, 5]. This approach firstly proposes points in objects instead of bounding-boxes of objects, then predicts a pixel-precise instance mask over an image based on each proposal point. Thus, a point proposal approach can deal with non-axis-aligned objects and objects sharing the same bounding-box. However, since the number of background pixels is much larger than the number of object pixels, it is likely that a predicted mask includes a few false positive pixels. For robot picking task, false positive pixels apart from an object are problematic. Therefore, we introduce rectangular masks to suppress those false positives. For each proposal point, our model generates a rectangular mask to restrict the existence area of the corresponding object within the rectangle.

Experimental results on WISDOM dataset [6] show that our method achieves superior performance to Mask R-CNN. In ablation study, introduction of rectangular masks gives a large improvement of box AP by 7.5 points and a small improvement of mask AP by 0.5 points.

2 Related Work

Two-stage methods with object detector/proposal [7, 8, 3] became the first mainstream of instance segmentation in deep learning. Mask R-CNN [3] is the most famous method and widely used even today. There are methods [9–11] to improve mask R-CNN performance. This two-stage approach firstly detects RoIs by using Region Proposal Network (RPN) [12] then classifies and segments an object in each RoI. Because this approach assumes that an RoI contains only one object, it cannot deal with objects sharing the same RoI. Recently, single-stage methods [13–16] based on the successful single-stage object detector FCOS [17] have been proposed and achieve comparable performance to mask R-CNN. However, these methods are still based on an object detector, hence they have the same problem as two-stage methods.

Another approach on instance segmentation is based on pixel embeddings [18, 19]. In this approach, the network is trained to output similar embeddings for the pixels of the same objects while dissimilar embeddings for the pixels belonging to different objects. Then, after outputting pixel embeddings, some clustering method is applied to them in order to obtain instance masks. This approach can produce a pixel-precise mask even for objects sharing the same bounding-box. However, this solution is suboptimal since the trained network is not optimized for instance segmentation. As a result, this approach is not competitive to the above mentioned detector based approach in terms of detection performance.

There are a few methods [4, 5] based on point proposal. These methods firstly propose points in objects, then predict an instance mask over an image for

each proposal point. Neven et al. [4] uses pixel embeddings for instance mask prediction. Therefore, this approach inherits both advantages of detector based approach and pixel embedding approach. Our method is based on this approach, and some techniques are introduced to improve the performance of instance segmentation of a top-view image.

Recently, a single stage method named SOLO [20] has been proposed. SOLO utilizes a uniform grids instead of proposal points. That is, it divides an image into grid cells. Then, for each grid cell, it predicts the mask of the object which the cell belongs to. Therefore, if the cell size is enough small, SOLO has the same advantages as point proposal approach.

3 Method

The overview of our instance segmentation method is shown in figure 1. Our model is built on a backbone such as ResNet [23] and FPN [21]. Multi-scale feature maps P2, P3, P4 and P5 are fused to a single scale shared features $\boldsymbol{f} \in \mathrm{R}^{H \times W \times 256}$ by the method described in [22] where H and W are height and width of P2, respectively. Then, shared features are input to five branches. Each branch has its own learnable parameters of two convolution layers. A point proposal branch outputs proposal points $\boldsymbol{p}$, and instance masks $\boldsymbol{M}$ are predicted by using pixel embeddings, scales, and proposal points. At the same time, rectangular masks $\boldsymbol{B}$ are generated by using an output of box branch and proposal points. Then, each instance mask is refined by the corresponding rectangular mask in order to suppress false positive pixels. Finally, confidence scores of instances are computed from the refined instance masks and a semantic segmentation score map.

3.1 Point Proposal

A point proposal branch transforms shared features to a heat map $\boldsymbol{h} \in \mathrm{R}^{H \times W}$ in which the pixels belonging to objects have larger values than background. Then, the at most K largest local maximum points $\{\boldsymbol{p}_k = (u_k, v_k)\}_{k=1}^{K}$ are sampled from the heat map. u_k and v_k represent y and x coordinates, respectively. These points $\{\boldsymbol{p}_k\}_{k=1}^{K}$ are used as proposal points. We use $K = 500$ in our experiments.

3.2 Mask Prediction

For each proposal point $\boldsymbol{p} = (u, v)$, its corresponding instance mask $\boldsymbol{M}_{\boldsymbol{p}} \in \{0, 1\}^{H \times W}$ is obtained by thresholding $\boldsymbol{m}_{\boldsymbol{p}} \in \mathrm{R}^{H \times W}$ whose elements are computed by the following function:

$$m_{\boldsymbol{p}}(i, j) = \exp\left(-(\boldsymbol{x}(i, j) - \boldsymbol{x}_{\boldsymbol{p}})^T \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} (\boldsymbol{x}(i, j) - \boldsymbol{x}_{\boldsymbol{p}})\right), \tag{1}$$

where $\boldsymbol{x}(i, j) \in \mathrm{R}^2$ is the pixel embedding at the coordinates (i, j) and $\boldsymbol{x}_{\boldsymbol{p}} = \boldsymbol{x}(u, v)$. $\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} \in \mathrm{R}^{2 \times 2}$ is a positive definite symmetric matrix at the proposal

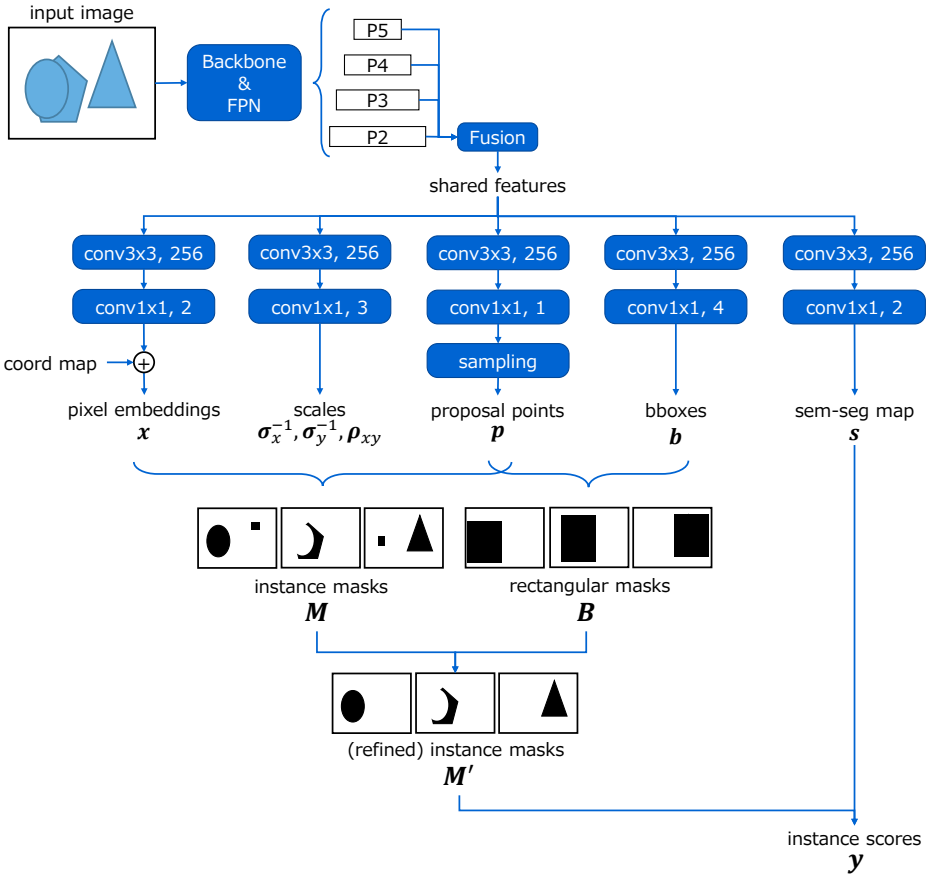

Fig. 1. The overview of our instance segmentation method. There are five branches after shared features, obtained by fusing P2, P3, P4 and P5 of FPN [21]. Fusion process is the same as UPerNet [22]. Instance masks are computed from pixel embeddings, scales, and proposal points, and rectangular masks are computed from bounding-boxes and proposal points, in parallel. Then, instance masks are refined by rectangular masks in order to suppress false positive pixels. Finally, confidence scores for instances are calculated from the refined masks and semantic segmentation map.

point $\boldsymbol{p}$ and used for distance calculation in the pixel embedding space. $m_{\boldsymbol{p}}(i,j)$ represents a probability that a pixel (i,j) belongs to the object which $\boldsymbol{p}$ belongs to. In order to make pixel embeddings $\boldsymbol{x}$ more discriminative, a coordinate map is added as mentioned in [24, 4]. Eq. (1) is similar to [4], but our $\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1}$ is a full matrix while their method adopts a diagonal one. We believe that a full matrix is more suitable for this task because objects are not axis-aligned. The matrix $\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1}$ is represented as:

$$\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} = \frac{1}{1-\rho_{xy,\boldsymbol{p}}^2} \begin{pmatrix} \sigma_{y,\boldsymbol{p}}^{-2} & -\rho_{xy,\boldsymbol{p}}\sigma_{x,\boldsymbol{p}}^{-1}\sigma_{y,\boldsymbol{p}}^{-1} \\ -\rho_{xy,\boldsymbol{p}}\sigma_{x,\boldsymbol{p}}^{-1}\sigma_{y,\boldsymbol{p}}^{-1} & \sigma_{x,\boldsymbol{p}}^{-2} \end{pmatrix}, \tag{2}$$

where

$$\boldsymbol{\Sigma}_{\boldsymbol{p}} = \begin{pmatrix} \sigma_{y,\boldsymbol{p}}^{2} & \rho_{xy,\boldsymbol{p}}\sigma_{x,\boldsymbol{p}}\sigma_{y,\boldsymbol{p}} \\ \rho_{xy,\boldsymbol{p}}\sigma_{x,\boldsymbol{p}}\sigma_{y,\boldsymbol{p}} & \sigma_{x,\boldsymbol{p}}^{2} \end{pmatrix} \tag{3}$$

$$\sigma_{y,\boldsymbol{p}}^{-1} = \sigma_{y}^{-1}(u,v) \tag{4}$$

$$\sigma_{x,\boldsymbol{p}}^{-1} = \sigma_{x}^{-1}(u,v) \tag{5}$$

$$\rho_{xy,\boldsymbol{p}} = \rho_{xy}(u,v). \tag{6}$$

Hence, our scales branch outputs three parameter maps $\boldsymbol{\sigma}_x^{-1}, \boldsymbol{\sigma}_y^{-1} \in \mathrm{R}_{>0}^{H\times W}$ and $\boldsymbol{\rho}_{xy} \in (-1,1)^{H\times W}$ for constructing each $\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1}$. The exponential activation function is used for $\boldsymbol{\sigma}_x^{-1}$ and $\boldsymbol{\sigma}_y^{-1}$, and the tanh activation function is used for $\boldsymbol{\rho}_{xy}$.

3.3 False Positive Suppression with Rectangular Masks

As mentioned above, since the majority of pixels are background, a few false positive pixels accidentally occur in background region. We suppress them in the instance mask $\boldsymbol{M}_{\boldsymbol{p}}$ by using a rectangular mask $\boldsymbol{B}_{\boldsymbol{p}} \in \{0,1\}^{H\times W}$:

$$\boldsymbol{M}'_{\boldsymbol{p}} = \boldsymbol{M}_{\boldsymbol{p}} \circ \boldsymbol{B}_{\boldsymbol{p}}, \tag{7}$$

where $\circ$ denotes Hadamard product. The rectangular mask $\boldsymbol{B}_{\boldsymbol{p}}$ is constructed from $\boldsymbol{b}_{\boldsymbol{p}} = (t_{\boldsymbol{p}}, l_{\boldsymbol{p}}, b_{\boldsymbol{p}}, r_{\boldsymbol{p}})^T \in \mathrm{R}_{>0}^4$, an output of our box branch at the proposal point $\boldsymbol{p} = (u,v)$, as follows:

$$B_{\boldsymbol{p}}(i,j) = \begin{cases} 1 & \text{if } u - \alpha t_{\boldsymbol{p}} \leq i \leq u + \alpha b_{\boldsymbol{p}} \wedge v - \alpha l_{\boldsymbol{p}} \leq j \leq v + \alpha r_{\boldsymbol{p}} \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

where α is a constant parameter to expand a predicted rectangle. We use $\alpha = 1.1$ in our experiments. The exponential activation function is applied after the last convolution layer in the box branch so that $\boldsymbol{b}_{\boldsymbol{p}}$ is positive. Note that this suppression procedure is not used in training phase.

Finally, an average of a semantic segmentation score map $\boldsymbol{s}$ over an instance region is used as a confidence score for the instance mask.

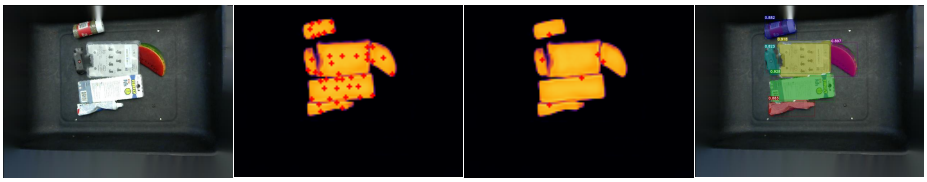

Fig. 2. Visualization of proposal points. (a) input image. (b) proposal points represented as red cross over proposal heat map $\boldsymbol{h}$. (c) remaining proposal points after post-processing. (d) instance segmentation result.

3.4 Post-processing

Our model outputs $\{(\boldsymbol{M}'_k, y_k)\}_{k=1}^{K}$ where $\boldsymbol{M}'_k$ is the (refined) instance mask obtained in eq. (7) and y_k is the corresponding confidence score. The obtained masks are generally redundant because there can be more than one proposal points in each object region. In order to remove redundant masks, we select the instance masks in descending order of confidence scores so that each instance mask includes only one proposal point. Then, redundant masks are removed from remaining masks based on overlap ratio. Figure 2 shows that an example of proposal points before and after post-processing. We can see that there is only one proposal point in each instance after post-processing.

3.5 Training

For training our model, we compute four losses $L_{\text{prop}}, L_{\text{mask}}, L_{\text{box}}$ and L_{sseg}. Then, these losses are combined by using multitask uncertainty weighting [25] which is one of the adaptive multitask loss balancing techniques. As described later, proposal points are required to compute L_{prop} and L_{mask}. However, the quality of point proposal is poor at an early stage of training. Therefore, we use points randomly sampled from ground-truth instance masks as proposal points during training. In our experiments, we sample 100 points per image. Each loss is briefly described below.

***L_{prop}* for Point Proposal.** We train the network so that heat map $\boldsymbol{h}$ can be used as a proxy to the quality of the instance mask generated at each proposal point. The quality at (i, j) is approximated with the following equation:

$$q(i,j) = \frac{1}{Z} \min_k m_{(i,j)}(u_k, v_k), \tag{9}$$

where (u_k, v_k) is the k-th random proposal point as described above and Z is a normalization parameter to ensure $\max_{i,j} q(i,j) = 1$. By using $q(i,j)$, we define L_{prop} as follows:

$$L_{\text{prop}} = \frac{1}{HW} \left| \boldsymbol{h} - \hat{\boldsymbol{h}} \right|_F^2, \tag{10}$$

where $|*|_F$ denotes Frobenius norm and

$$\hat{h}(i,j) = \begin{cases} \beta + (1-\beta)q(i,j) & \text{if } (i,j) \in \text{foreground pixels} \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

$\beta \in (0,1)$ is a minimum value for foreground pixels and we set it to 0.7. This value is the same as [5].

L_{mask} for Mask Prediction. For L_{mask}, we adopt a simple soft-IoU loss [26] as follows:

$$L_{\text{mask}} = \frac{1}{K}\sum_{k=1}^{K}\left(1 - \frac{\epsilon + \sum_{i,j} m_{\boldsymbol{p}_k}(i,j)\hat{M}_{\boldsymbol{p}_k}(i,j)}{\epsilon + \sum_{i,j} m_{\boldsymbol{p}_k}(i,j) + \hat{M}_{\boldsymbol{p}_k}(i,j) - m_{\boldsymbol{p}_k}(i,j)\hat{M}_{\boldsymbol{p}_k}(i,j)}\right), \tag{12}$$

where $\epsilon = 1$ is a smoothing constant and $\hat{\boldsymbol{M}}_{\boldsymbol{p}_k}$ is the ground-truth instance mask to which the point $\boldsymbol{p}_k$ belongs.

L_{box} for Rectangular Masks. For L_{box}, we compute box IoU loss [27] at every foreground pixels as below:

$$L_{\text{box}} = \frac{1}{N}\sum_{\boldsymbol{p}\in\text{foreground}}\left(1 - \frac{I_{\boldsymbol{p}} + \epsilon}{U_{\boldsymbol{p}} + \epsilon}\right), \tag{13}$$

where N is the number of foreground pixels, $\epsilon = 1$ is a smoothing constant,

$$I_{\boldsymbol{p}} = \left(\min(t_{\boldsymbol{p}}, \hat{t}_{\boldsymbol{p}}) + \min(b_{\boldsymbol{p}}, \hat{b}_{\boldsymbol{p}})\right)\left(\min(l_{\boldsymbol{p}}, \hat{l}_{\boldsymbol{p}}) + \min(r_{\boldsymbol{p}}, \hat{r}_{\boldsymbol{p}})\right), \tag{14}$$

$$U_{\boldsymbol{p}} = (\hat{t}_{\boldsymbol{p}} + \hat{b}_{\boldsymbol{p}})(\hat{l}_{\boldsymbol{p}} + \hat{r}_{\boldsymbol{p}}) + (t_{\boldsymbol{p}} + b_{\boldsymbol{p}})(l_{\boldsymbol{p}} + r_{\boldsymbol{p}}) - I_{\boldsymbol{p}}, \tag{15}$$

and $\hat{*}$ denotes the corresponding ground-truth, respectively.

L_{sseg} for Semantic Segmentation. For the semantic segmentation loss L_{sseg}, we adopt a standard soft-max cross entropy loss with label smoothing [28]. Its smoothing parameter is set to 0.1 in our experiments.

4 Experiments

The performance of our method is evaluated on WISDOM Dataset [6]. This dataset provides 400 color top-view images of $1,032 \times 772$. We use the same training/test split as [6]. This split provides 100 training images including 25 objects and 300 test images including different 25 objects. For all experiments, each input image is resized so that the longer side is equal to 512. The detection performance is evaluated three times with different random seeds, and their averaged score is reported.

Here, we consider evaluation metrics. As described above, false positive pixels apart from an object are problematic for robot picking task, hence an evaluation metric should be sensitive to them. Box AP, an object detection evaluation metric, is such a metric while mask AP, commonly used as an instance segmentation

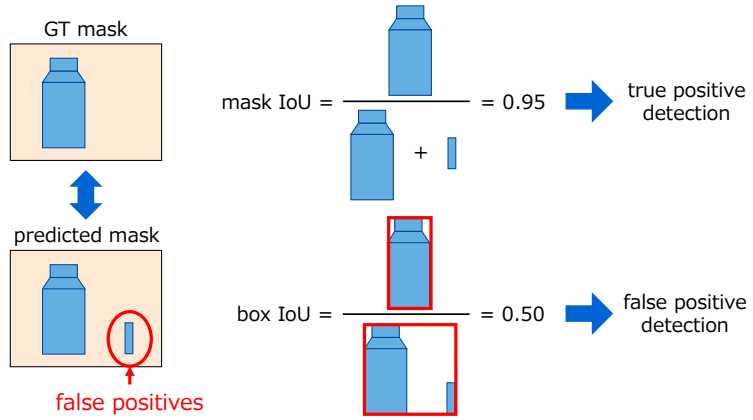

Fig. 3. Comparison of mask IoU and box IoU. A predicted mask including a few false positive pixels apart from an object is judged as true positive detection by mask IoU while judged as false positive detection by box IoU under an IoU threshold in $(0.5, 0.95]$.

evaluation metric, is insensitive to a few false positive pixels. Figure 3 shows a typical example. A predicted mask including a few false positives is judged as true positive by mask IoU while judged as false positive by box IoU. On the other hand, box AP is insensitive to mask accuracy while mask AP is sensitive to it. Therefore, we use both evaluation metrics box AP and mask AP in our experiments.

We use ResNet50 model pre-trained on ImageNet [29]. Our model is trained for $10,000$ iterations with a batch-size of 4. We use the SGD optimizer with learning rate of 0.02, momentum of 0.9 and weight decay of 0.0005. The learning rate is scheduled using a cosine annealing scheduler [30]. During training, parameters of stem and stage1 in ResNet50 are fixed and the learning rate for the other parameters in ResNet50 is multiplied by 0.1. We apply data augmentation similar to that used to train an SSD model [31]. Data augmentation is implemented by using Albumentations library [32]. We use PyTorch framework [33] for all our experiments.

4.1 Main Results

Table 1 shows comparison of our model with other methods. The evaluation result of Mask R-CNN with ResNet50-FPN is obtained by using maskrcnn-benchmark [34]. Our model achieves mask AP of 52.3% and box AP of 48.1%, which are 12.2 points and 11.4 points higher than those of Mask R-CNN with the same ResNet50-FPN backbone, respectively. As compared with D-SOLO [20], our mask and box APs are 10.3 and 9.0 points higher, respectively. Furthermore, our method is comparable to SD Mask R-CNN[1] [6] which uses depth information.

[1] Since their ResNet35 model is not standard, we could not compare the performance with the same backbone. The configuration of ResNet35 can be found in their code.

Fig. 4. Visualization results of our method, Mask R-CNN and D-SOLO.

Table 1. Mask AP and box AP of each method on WISDOM dataset.

method	backbone	input type	mask AP @all [%]	box AP @all [%]
SD Mask R-CNN [6]	ResNet35-FPN	depth	51.6	-
Mask R-CNN [6]	ResNet35-FPN	RGB	38.4	-
Mask R-CNN	ResNet50-FPN	RGB	40.1	36.7
D-SOLO	ResNet50-FPN	RGB	42.0	39.1
ours	ResNet50-FPN	RGB	52.3	48.1

Table 2. Mask APs [%] of each method with ResNet50-FPN backbone on WISDOM dataset.

method	AP@all	AP@IoU=0.5	AP@IoU=0.75
Mask R-CNN	40.1	76.4	38.0
D-SOLO	42.0	75.1	42.9
ours	52.3	82.8	55.1

Table 2 shows mask APs at IoU threshold of 0.5 and 0.75. We can see that mask AP@IoU=0.75 of our method is much higher than mask R-CNN and D-SOLO. This means that our method can predict object shapes more precisely than those methods. We show some visualization results of each method in figure 4. It also shows that our method produces more accurate instance masks than Mask R-CNN and D-SOLO. Moreover, our method generates good instance segmentation results even for highly overlapped objects. Typical failure case is found in the bottom-most row in figure 4. All the methods over-segment a hammer-shaped object. This is because there is no object of similar shape in training data.

4.2 Ablation Study

The results of ablation experiments are shown in table 3. Our instance segmentation method with rectangular masks achieves a large improvement of box AP from 40.6% to 48.1% while a small improvement of mask AP from 50.7% to 52.3%. This means that introduction of rectangular masks successfully suppress false positive pixels apart from objects. An example is shown in figure 5. We can

Table 3. Ablation experiments on WISDOM dataset.

full $\boldsymbol{\Sigma}$	rectangular masks	mask AP@all [%]	box AP@all [%]
		49.2	39.1
✓		50.7	40.6
	✓	51.8	47.4
✓	✓	52.3	48.1

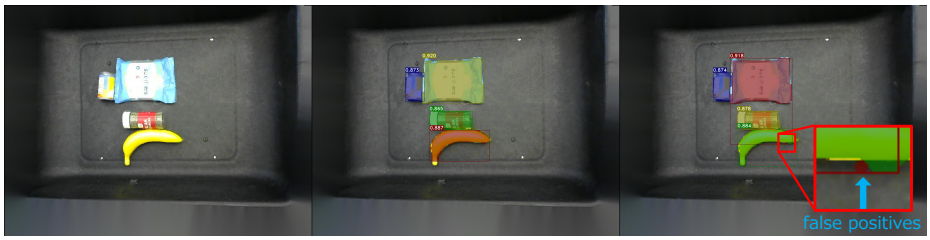

Fig. 5. Instance masks obtained by our method with or without rectangular masks. (a) Input image. (b) Instance masks with rectangular masks. (c) Instance masks without rectangular masks. The instance mask for upper-right object includes a few false positives under the banana-shaped object.

see that our method without rectangular masks produces a few false positive pixels far from an object while our method with rectangular masks does not produces such a false positive pixels.

By using a full matrix $\boldsymbol{\Sigma}$, the performance mask AP is slightly improved from 51.8% to 52.3%. Computational cost is almost unchanged between our model with a full matrix and that with diagonal one. Hence, introducing a full matrix is a good choice for this task.

5 Conclusions

In this work, we focus on instance segmentation of a top-view image for robot picking task. We propose a point proposal based instance segmentation method with rectangular masks, which suppress false positive pixels apart from objects. The experimental results on WISDOM dataset show that our method achieves superior performance to Mask R-CNN and D-SOLO with the same backbone model.

References

1. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, L.: Microsoft coco: Common objects in context. In: The European Conference on Computer Vision (ECCV). (2014)
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
3. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: The IEEE International Conference on Computer Vision (ICCV). (2017)
4. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)

5. Sofiiuk, K., Barinova, O., Konushin, A.: Adaptis: Adaptive instance selection network. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
6. Danielczuk, M., Matl, M., Gupta, S., Li, A., Lee, A., Mahler, J., Goldberg, K.: Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data. In: The IEEE International Conference on Robotics and Automation (ICRA). (2019)
7. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
8. Viet Pham, S.I., Kozakaya, T.: Biseg: Simultaneous instance segmentation and semantic segmentation with fully convolutional networks. In: The British Machine Vision Conference (BMVC). (2017)
9. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
10. Cheng, T., Wang, X., Huang, L., Liu, W.: Boundary-preserving mask r-cnn. In: ECCV. (2020)
11. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
12. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., eds.: Advances in Neural Information Processing Systems 28. Curran Associates, Inc. (2015) 91–99
13. Ying, H., Huang, Z., Liu, S., Shao, T., Zhou, K.: Embedmask: Embedding coupling for one-stage instance segmentation (2019)
14. Wang, Y., Xu, Z., Shen, H., Cheng, B., Yang, L.: Centermask: Single shot instance segmentation with point representation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
15. Lee, Y., Park, J.: Centermask: Real-time anchor-free instance segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
16. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: The European Conference on Computer Vision (ECCV). (2020)
17. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
18. Fathi, A., Wojna, Z., Rathod, V., Wang, P., Song, H.O., Guadarrama, S., Murphy, K.P.: Semantic instance segmentation via deep metric learning. CoRR **abs/1703.10277** (2017)
19. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation for autonomous driving. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. (2017)
20. Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L.: SOLO: Segmenting objects by locations. In: The European Conference on Computer Vision (ECCV). (2020)
21. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
22. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: The European Conference on Computer Vision (ECCV). (2018)

23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
24. Novotny, D., Albanie, S., Larlus, D., Vedaldi, A.: Semi-convolutional operators for instance segmentation. In: The European Conference on Computer Vision (ECCV). (2018)
25. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
26. Rahman, M.A., Wang, Y.: Optimizing intersection-over-union in deep neural networks for image segmentation. In: International Symposium on Visual Computing. (2016) 234–244
27. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115** (2015) 211–252
30. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (ICLR) 2017 Conference Track. (2017)
31. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision – ECCV 2016, Springer International Publishing (2016) 21–37
32. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: Fast and flexible image augmentations. Information **11** (2020)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32. Curran Associates, Inc. (2019) 8024–8035
34. Massa, F., Girshick, R.: maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark (2018)