Multi-level Prediction for Overlapped Parcel Segmentation

Zhequan Zhou[1(✉)], Shujing Lyu[2,3(✉)], and Yue Lu[2,3(✉)]

[1] School of Computer Science and Technology, East China Normal University, Shanghai, China
zqzhou@stu.ecnu.edu.cn

[2] School of Communication and Electronic Engineering, East China Normal University, Shanghai, China
{sjlv,ylu}@cs.ecnu.edu.cn

[3] Shanghai Key Laboratory of Multidimensional Information Processing, Shanghai, China

Abstract. In this paper, we propose a new instance segmentation framework based on a multi-level prediction mechanism, aiming at segmenting overlapped parcels. In this framework, one location on FPN's feature maps can predict a set of overlapped instances through a multi-level head architecture according to their overlapping order. Besides, to avoid the inherent limit of bounding boxes in object overlapping scenes, our approach is bounding-box free. And we also provide a dataset for overlapped parcel instance segmentation named OLParcel. On a Mask RCNN baseline, our network can improve 4.25% AP, 4.26% recall, and 7.11% MR^{-2} on our dataset.

Keywords: Instance segmentation · Parcel sorting · Overlap

1 Introduction

Lots of advanced instance segmentation methods have been proposed [2,6,12,15,23,25,27] and have achieved remarkable performances on many popular datasets, such as COCO [14] and PASCAL VOC [5]. However, they still have limitations in some stacked or overlapped scenarios. Such as in parcel sorting application, parcels on the conveyor belt are randomly stacked and highly overlapped with each other. As a result, these advanced instance segmentation frameworks still face challenges when predicting these parcels. Figure 1 shows a typical failure case in this scene: Mask RCNN [6] fails to predict a parcel highly overlapped with others (indicated by a red dash box). This kind of failure is mainly ascribed to two reasons. The first reason is that bounding boxes aren't suitable for these scenes, since highly overlapped objects are likely to have highly overlapped bounding boxes. Therefore, it is difficult for a network to generate distinguishing prediction for each proposal. Besides, these predictions are prone to be mistakenly suppressed by box-based Non-maximum Suppression (NMS). The second reason is the lack of a special mechanism to explicitly handle this overlapping situation.

G. Zhai et al. (Eds.): IFTC 2021, CCIS 1560, pp. 3–17, 2022.
https://doi.org/10.1007/978-981-19-2266-4_1

For example, when Mask RCNN tries to predict these parcels in Fig. 1(a), there is an intractable ambiguity: it is not clear w.r.t which parcel to be predicted in these overlapped regions.

(a) Input image (b) Results

Fig. 1. Instance segmentation results of baseline (Mask RCNN [6]).

Some works have tried to address this issue in objects overlapping scenes from several different perspectives, such as new loss functions [26,29] and sophisticated NMS [1,9–11,16]. However, as we will analyze later (Sect. 2), since these proposed methods still base on bounding boxes, it's still difficult for them to distinguish highly overlapped parcels like in Fig. 1. Therefore, to avoid the limit of bounding boxes in object overlapping scenes, our approach is bounding-box free. Further, to resolve the prediction ambiguity in overlapping regions, a multi-level prediction mechanism is proposed to explicitly handle overlapping situation based on parcels' overlapping order.

Besides, we also contribute an overlapped parcel instance segmentation dataset, named OLParcel, which contains 1,539 images with 8,631 parcels in them. Section 4.1 shows more information about OLParcel.

2 Related Work

2.1 Bounding Box in Instance Segmentation

In the instance segmentation task, many methods use bounding boxes as an intermediate representation before generating final masks.

Bounding Box in Generating Proposals. Proposal-based instance segmentation frameworks [6,12,15] need first generate lots of proposals. These region proposals can be regarded as a large set of bounding boxes spanning the full image. However, the overlap of objects increases the difficult in object localization when generating proposals. Some object detection works [26,29] designed new loss functions to address this problem. They enforce proposals to be close to their corresponding ground truths or introduce extra penalties to push proposals away from other ground truths. The quality of detections is improved with the help of these new loss functions. However, since box-based NMS is still used in these frameworks, the issue that overlapped objects may be mistakenly suppressed remains unresolved.

Bounding Box in Generating Masks. Many instance segmentation frameworks like Mask RCNN and its variants [6,12,15] generate final masks by performing binary segmentation inside their proposals. This kind of instance segmentation methods is called box-based method. However, since highly overlapped instances have highly overlapped proposals, it is difficult for them to generate distinguishing predictions for these proposals.

Bounding Box in Post-processing. The effectiveness of naïve box-based NMS is based on the assumption that multiple instances rarely occur at the same location, which is no longer satisfied in the object overlapping scenes. Many improved NMS strategies have been proposed to resolve this issue. For example, Soft-NMS [1] and Softer-NMS [8] suggest decaying the confidence score of the neighboring predictions rather than directly discard them. Jan et al. [10] and Lu et al. [18] proposed a new neural network to perform NMS. Other works such as Tnet [9] and AdaptiveNMS [16], proposed to predict different NMS thresholds for different bounding boxes. Although these works improve the detection performance in object overlapping scenes, it is still difficult to distinguish highly overlapped objects as in Fig. 1 due to their highly overlapped bounding boxes.

In conclusion, based on the above analyses, we argue that one key issue in object overlapping scenes lies in the inherent limit of bounding boxes: highly-overlapped objects always have highly-overlapped bounding boxes. Therefore, we try to build a bounding-box free framework via building our approach based on box-free methods like CondInst [23] and using mask-based NMS in post-processing. Instead of encoding instances into bounding boxes, these box-free methods encode them into "mask coefficients" [2] or "generating filters" [23], which is a much more flexible and accurate way.

2.2 Multiple Instance Prediction

In this paper, we propose a multi-level prediction mechanism to predict a set of overlapped parcels. This idea of multiple instance prediction is not totally new. Some previous works have implied the idea of multiple instance prediction.

Multiple Instance Prediction via Different Object Aspect Ratios. Some networks [6,17,19,20] set various anchor boxes of different aspect ratios at each location. These anchor boxes can be viewed as pre-defined proposals or sliding windows. As a result, this kind of method can predict multiple objects of different aspect ratios at a location with these anchor boxes. However, these pre-defined anchor boxes result in many hyper-parameters. The tuning of these hyper-parameters is very tricky.

Multiple Instance Prediction via Different Object Sizes. Most instance segmentation frameworks have been equipped with feature pyramid network (FPN) [13] to predict objects of different sizes. Therefore, the overlaps caused by different object sizes have been alleviated with this multi-level prediction on many datasets such as COCO [14] and PASCAL VOC [5]. However, in single-class

datasets like our OLParcel, CityPersons [28] or CrowdHuman [22], when objects are in the same category and have similar size, it is unpractical to resolve overlapping situations only through FPN.

In conclusion, we argue that it's difficult to distinguish overlapped objects only through their sizes and aspect ratios. It inspires us to design a special mechanism to explicitly handle this overlapping situation according to the characteristics of overlapped parcels, such as overlapping order.

3 Our Approach: Multi-level Prediction

We propose a multi-level prediction mechanism based on parcels' overlapping order to resolve the ambiguity about which parcel is to be predicted in the overlapped regions. Parcel Level (PL for short) is defined to describe the overlapping order of parcels in this paper. Furthermore, to predict parcels of different PLs, we extend output heads of the instance segmentation framework to multi-level heads.

The details of our approach are introduced as follows.

3.1 Parcel Level

Algorithm 1. Compute occlusion relations of two parcels

Input:
M_i is the i-th parcel's mask
M_j is the j-th parcel's mask
Output:
$\mathcal{R}$ is a integer representing for occlusion relations between parcel i and parcel j.

compute Convex full of M_i: $\mathcal{C}_i = Convex(M_i)$
compute Convex full of M_j: $\mathcal{C}_j = Convex(M_j)$
compute Intersection over Union between $\mathcal{C}_i$ and $\mathcal{C}_j$: $U_{ij} = \mathcal{C}_i \cap \mathcal{C}_j$
if $U_{ij} < thr$ **then**
  $\mathcal{R} = 0$
else
  compute intersection area between $\mathcal{C}_i$ and $\mathcal{C}_j$: $I_{ij} = \mathcal{C}_i \cap \mathcal{C}_j$
  compute intersection area between I_{ij} and M_i: $\mathcal{A}_i = Area(I_{ij}, M_i)$
  compute intersection area between I_{ij} and M_j: $\mathcal{A}_j = Area(I_{ij}, M_j)$
  if $\mathcal{A}_i < \mathcal{A}_j$ **then**
    $\mathcal{R} = 1$
  else
    $\mathcal{R} = 0$
  end if
end if
return $\mathcal{R}$

Parcel Level (PL) represents the top-down order of a parcel in overlapped scene. In an image, $PL(i,(x,y))$ means the level of i-th parcel at image location (x,y). It can be computed based on the ground-truth masks of the image. To obtain a parcel's PL, we should know how many parcels are stacked upon it. We decompose this problem into sub problems that how to compute occlusion relations of any two overlapped parcels, namely which one is occluder and which one is occludee. We approximately consider the convex hull calculated by a parcel's visible parts as its non-occluded shape. In the rest paper, we use M_i to denote the ground-truth mask of i-th parcel and $\mathcal{C}_i$ to denote the convex hull calculated by M_i. We design Algorithm 1 to compute occlusion relations of two parcels.

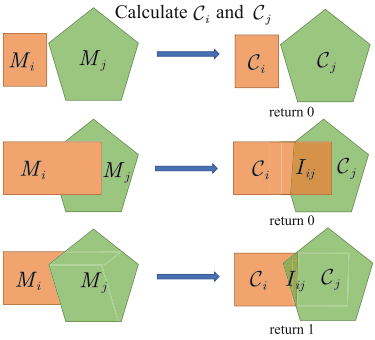
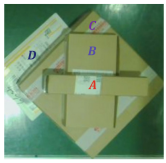
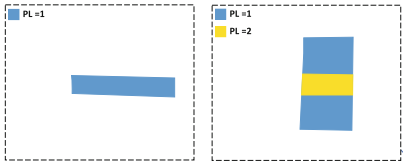


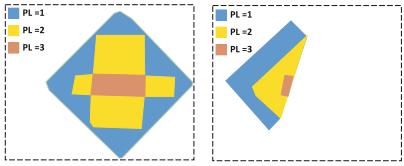
Fig. 2. Algorithm 1.



(a)



(b) PL of parcel A (c) PL of parcel B



(d) PL of parcel C (e) PL of parcel D

Fig. 3. The examples of Parcel Level.

The key idea of Algorithm 1 is shown in Fig. 2: for i-th parcel and j-th parcel, we first calculate their convex hulls $\mathcal{C}_i$ and $\mathcal{C}_j$; if the Intersection over Union (IoU) between $\mathcal{C}_i$ and $\mathcal{C}_j$ is larger than a threshold (0.05 by default), we consider these two parcels to be overlapped with each other, otherwise the algorithm returns 0 (the first row in Fig. 2); then we calculate the intersection area I_{ij} of $\mathcal{C}_i$ and $\mathcal{C}_j$; if j-th parcel is an occludee, the I_{ij} should be in M_i and not in M_j, and the algorithm also returns 0 (the second row in Fig. 2); if i-th parcel is an occludee, the algorithm returns 1 (the third row in Fig. 2). We label Algorithm 1 as $\mathcal{S}$.

Formally, the PL of i-th parcel at image location (x, y) can be formulated as,

$$PL(i,(x,y)) = \begin{cases} 1 + \sum_{j}^{j \neq i} \mathcal{I}(\mathcal{C}_j,(x,y)) * \mathcal{S}(M_i, M_j), & (x,y) \in \mathcal{C}_i \\ 0, & (x,y) \notin \mathcal{C}_i \end{cases} \quad (1)$$

where $\mathcal{I}$ is a function judging if a location (x, y) is in a parcel's convex hull; $\mathcal{I}$ returns 1 when the location is in the convex hull of this parcel, otherwise returns 0; $\mathcal{S}$ is Algorithm 1. In training phase, we use the PL labels generated by Algorithm 1 and Eq. 1 to train our network.

Figure 3 shows examples of PL. In Fig. 3(a), there are four parcels: parcel A, parcel B, parcel C, and parcel D. The remains show these four parcels' PLs respectively.

3.2 Multi-level Head Architecture

In previous box-free instance segmentation frameworks like SOLOv2 [27] and CondInst [23], on each FPN's feature maps P_i, several output heads, such as Classification Head and Controller Head, are applied to make instance-related predictions. Each output head includes one branch. Therefore, for each location on P_i, these heads output at most one prediction. However, in our approach, each location on P_i can be associated with multiple overlapped parcels, so we extend these output heads to multi-level heads through appending extra parallel $K-1$ branches to each original head where K is a given constant standing for the maximum PL we predict. Each multi-level head includes K parallel branches, and for a location (x, y) on P_i, the k-th branch tries to predict a parcel whose PL is k. As a result, for each location on P_i, branch 1 to K can predict at most K parcels.

3.3 Network Architecture

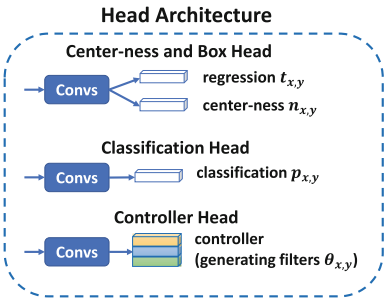


Fig. 4. Head architecture of CondInst.

We choose CondInst [23] as a baseline network to build our framework. Figure 4 shows the head architecture of CondInst. In CondInst, there are three output

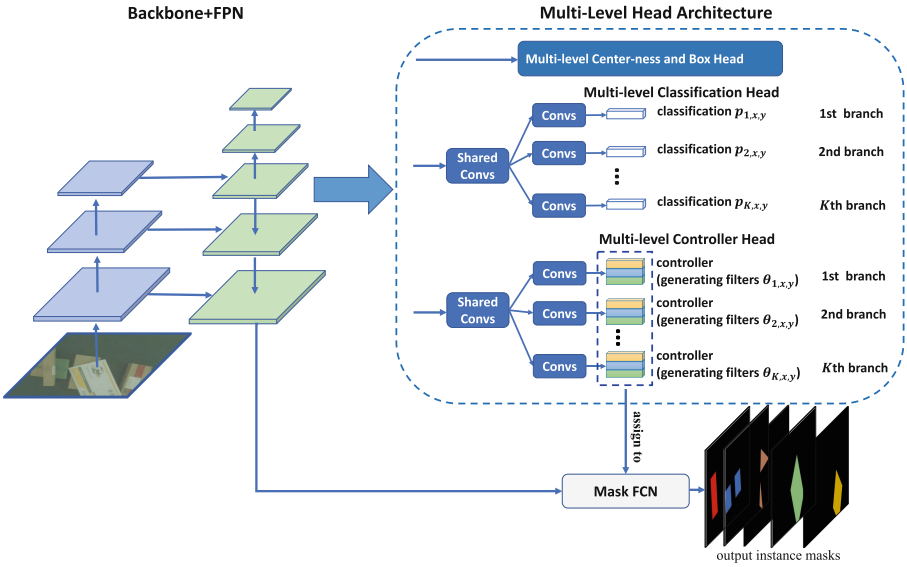


Fig. 5. Network architecture.

heads: Center-ness and Box Head, Classification Head, and Controller Head. Our framework is achieved by extending these output heads to multi-level heads. Figure 5 illustrates our method based on CondInst. The multi-level head architecture in the upper right corner of Fig. 5 includes three multi-level heads: Multi-level Center-ness and Box Head, Multi-level Classification Head, and Multi-level Controller Head (Multi-level Center-ness and Box Head is folded in this figure for the limitation of space). Each multi-level head includes K parallel branches. Following CondInst, each branch is composed of four 3×3 convolutions and a final output layer. Since the parameters in multi-level heads are K times compared to original output heads, we share the first few layers among K branches in each multi-level head to make network parameter-efficient. In our paper, the first one layer is shared on the trade-off between accuracy and computational cost.

For an input image $I \in \mathbb{R}^{H \times W \times 3}$, the goal of our work is to predict the pixel-level mask for each parcel. The ground-truths in our work are defined as $\{(M_i, \mathcal{K}_i)\}$, where $M_i \in \{0,1\}^{H \times W}$ is the mask for i-th parcel and $\mathcal{K}_i \in \{0, \ldots, K\}^{H \times W}$ is this parcel's PL.

In CondInst, each location on the FPN's feature maps P_i either is associated with a parcel, thus being a positive sample, or is considered a negative sample. In our approach, each location can be associated with multiple parcels. More specifically, let us consider the feature maps $P_i \in \mathbb{R}^{H \times W \times C}$. As shown in previous works, a location (x, y) on the feature maps can be mapped back onto the image as $\left(\lfloor \frac{s}{2} \rfloor + xs, \lfloor \frac{s}{2} \rfloor + ys\right)$ where s is down-sampling ratio of P_i. If the mapped location falls in the center regions of some parcels, the location is con-

sidered to be responsible for predicting them. Then, for a parcel whose PL is k at $\left(\lfloor \frac{s}{2} \rfloor + xs, \lfloor \frac{s}{2} \rfloor + ys\right)$, the k-th branch is responsible for predicting this parcel. A parcel's center region is defined as the box $(c_x - rs, c_y - rs, c_x + rs, c_y + rs)$, where (c_x, c_y) denotes the mass center of the parcel and r is a constant scalar being 1.5 as in CondInst. Other architectures such as the backbone and feature pyramid network are the same as CondInst.

3.4 Loss Function

Formally, the overall loss function of our approach can be formulated as,

$$L_{overall} = \frac{1}{K} \sum_{k=1}^{K} L^k_{output} + \lambda L_{mask}, \tag{2}$$

where L^k_{output} and L_{mask} denote the loss of k-th branch's outputs and the original loss for instance masks in CondInst, respectively. λ being 1 in this work is used to balance the two losses. As CondInst bases on FCOS [24], $\sum_{k=1}^{K} L^k_{output}$ can be considered as the sum of K FCOS losses, and each L^k_{output} is the same as the L_{fcos} in FCOS paper.

3.5 Inference

In inference, given an input image, we forward it through the network to obtain outputs including classification confidence $p_{k,x,y}$, center-ness scores $n_{k,x,y}$, box prediction $t_{k,x,y}$, and generated filters $\theta_{k,x,y}$ where k is represent for k-th branch. We first follow steps in CondInst to obtain new classification score $p_{k,x,y}$ by multiplying original classification confidence $p_{k,x,y}$ with the corresponding center-ness score. Then we use a confidence threshold of 0.05 to filter out predictions with low confidence. Afterwards, for each PL k, the top 100 predictions outputted by k-th branch are used to compute masks. The steps computing masks are the same as CondInst. As a result, there are totally 100K predictions to be used to compute masks. In post-processing, we choose Matrix-NMS proposed in SOLOv2 [27], which based on predicted masks and has similar performance but faster compared to naïve mask-based NMS. These 100k computed masks are sent to Matrix-NMS, and re-scored $p_{k,x,y}$ is obtained. Finally, we choose the masks with $p_{k,x,y} > 0.05$ as final predictions.

4 Experiments

In this section, we evaluate our method from different perspectives.

4.1 Dataset: OLParcel

In this study, we collect a parcel dataset in overlapped scene and name it OLParcel.

The OLParcel dataset contains a total of 1,539 images, including 8,631 parcels. All images are stored in PNG format with the resolution of 360×954. OLParcel is partitioned into a training set and a testing set, the former containing 860 images with 4,856 parcels, and the latter containing 679 images with 3,775 parcels.

Table 1 lists the "Parcel Level Density" about OLParcel. The PLs of only 0.49% parcels are bigger than 2. So we set the max PL as 2, that is $K = 2$, in our experiments.

Table 1. Parcel Level Density in OLParcel

OLParcel	$PL = 1$	$PL = 2$	$PL \geq 3$
Training	87.93%	11.64%	0.43%
Testing	86.33%	13.11%	0.56%
Total	87.23%	12.28%	0.49%

4.2 Evaluation

Following previous works [3,16,22], AP, $recall$ and MR^{-2} [4] are used to evaluate network performances. Besides, for evaluating networks' performance on overlapped parcels more effectively, we design following two additional criterion:

- Overlapped Parcel Average Precision (AP_{OL}), which is only for overlapped parcels. For each prediction, we first calculate IoU with all ground-truths. Then, if the ground-truth having max IoU with this prediction is overlapped, we consider this prediction is responsible for predicting this overlapped parcel and call it overlapped prediction. Finally, we obtain AP_{OL} only using overlapped ground-truths and overlapped predictions.
- Overlapped Parcel Recall ($Recall_{OL}$), which is similar as AP_{OL}. We obtain $Recall_{OL}$ only using overlapped ground-truths and overlapped predictions.

4.3 Experiments on OLParcel

Implementation Details. We employ standard ResNet-50 [7] as the backbone and train it with image scale (shorter side) randomly sampled from [288, 448], which reduces overfitting. Left-right flipping data augmentation is also used during training. Inference is on a single scale of 360 pixels. We use a mini-batch size of 8 images on one GPU and train the model for 20k iterations. The initial learning rate is 0.005 and is decreased by 10 at the 10k iteration and 15k iteration. The backbone is initialized from the pre-trained model on ImageNet [21] classification. Other network settings is the same as original paper [23]. All naïve box-based NMS overlap IoU thresholds are set to 0.6 by default.

Comparison to Baselines. In this experiment, we compare the performance of our method with CondInst [23] and Mask RCNN [6], which are representative works categorized into box-free method and box-based method, respectively. For CondInst, we use the official open-source implementation in AdelaiDet[1]. And for Mask RCNN, we use the official open-source implementation in detectron2[2]. The initial learning rates of CondInst and Mask RCNN are 0.005 and 0.01 respectively. Other settings are the same as our approach.

Table 2. Comparisons of different methods using naïve box-based NMS with different IoU thresholds

IoU*	Method	AP	AP_{OL}	$Recall$	$Recall_{OL}$	MR^{-2}
0.5	Mask RCNN [6]	91.27	86.90	92.13	88.75	17.71
	CondInst [23]	90.36	87.04	91.36	88.15	18.57
	Ours	92.25	89.10	93.22	90.07	15.93
0.6	Mask RCNN [6]	92.21	89.58	93.75	91.22	16.79
	CondInst [23]	92.17	88.86	93.27	90.98	15.96
	Ours	94.02	92.46	95.50	94.09	**14.66**
0.7	Mask RCNN [6]	92.91	90.12	94.65	92.56	17.83
	CondInst [23]	92.90	90.96	94.70	93.04	18.51
	Ours	94.72	93.16	96.66	95.75	15.09
0.8	Mask RCNN [6]	93.14	90.45	95.60	94.01	20.86
	CondInst [23]	93.18	90.75	95.79	94.66	21.55
	Ours	**95.02**	**93.61**	**97.54**	97.05	16.91
0.9	Mask RCNN [6]	91.38	86.74	96.11	94.78	30.14
	CondInst [23]	92.39	88.86	96.48	95.71	26.42
	Ours	93.27	90.86	97.83	**97.49**	23.35

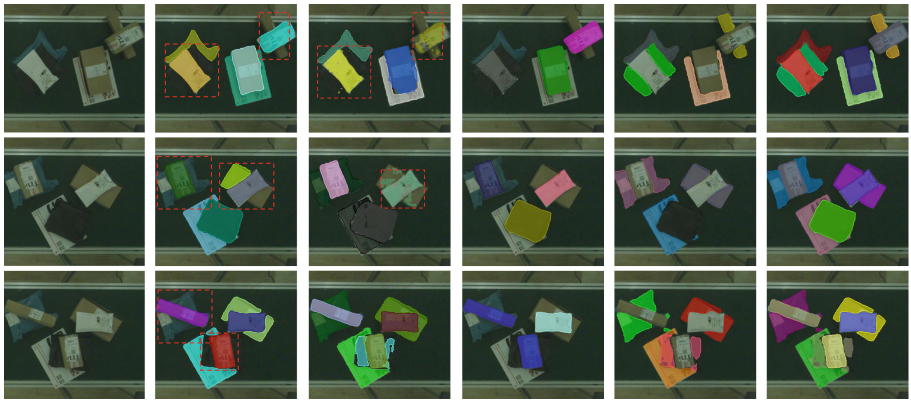
Table 2 shows comparisons of different methods using naïve box-based NMS with different IoU thresholds. We observe that at the same IoU threshold, our method improves about 1–2% in AP, 1–2% in $recall$, and 3–7% in MR^{-2} compared to CondInst and Mask RCNN. And for more challenge metrics AP_{OL} and $recall_{OL}$, our method obtains 2–3% and 2–4% gains respectively, which indicate the effectiveness of our multi-level prediction mechanism designed for segmenting overlapped parcels. We also observe that compared with default IoU threshold setting (0.6), slightly enlarging the IoU threshold (from 0.6 to 0.8) may help to recall more instances, so AP and $recall$ increase slightly; however, the MR^{-2} index suffers from significantly drops, indicating that high IoU threshold introduces more false predictions with high confidences.

[1] https://github.com/aim-uofa/AdelaiDet.
[2] https://github.com/facebookresearch/detectron2.

Table 3. Comparisons of different NMS strategies

NMS	Method	AP	AP_{OL}	$Recall$	$Recall_{OL}$	MR^{-2}
Box-NMS	Mask RCNN [6]	92.21	89.58	93.75	91.22	16.79
	CondInst [23]	92.17	88.86	93.27	90.98	15.96
	Ours	94.02	92.46	95.50	94.09	14.66
Matrix-NMS	Mask RCNN [6]	91.12	89.34	91.54	90.65	16.94
	CondInst [23]	91.49	88.23	92.53	89.81	16.01
	Ours	**96.46**	**95.93**	**98.01**	**97.73**	**9.68**

Table 3 lists the comparison results of different methods with different NMS strategies. The Box-NMS refers to naïve box-based NMS. Compared to use Box-NMS in post-processing, our Matrix-NMS based method improves 2.44% in AP, 2.51% in $recall$, and 4.98% in MR^{-2}, validating the effectiveness of discarding bounding boxes. Surprisingly, for CondInst and Mask RCNN, their performances are slightly dropped when using Matrix-NMS, which suggests that without suitable mechanism designed for overlapped objects, only using mask-base NMS isn't beneficial to performances.



(a) Input (b) CondInst (c) Mask RCNN(d) Ours: PL=1(e) Ours: PL=2 (f) Ours

Fig. 6. Visual comparison of the baselines and our approach.

Figure 6 shows visual results of baseline models and our approach. The first column are input images. The second column are the CondInst's results. The third column are the Mask RCNN's results. The fourth column are our approach's results produced by $1st$ branch. The fifth column are our approach's results produced by $2nd$ branch. The last column are our approach' results. The score threshold for visualization is 0.3. The failure cases are indicated in red dash boxes. As shown in Fig. 6(b) and 6(c), the baseline models face challenges

on predicting overlapped parcels, but our method still works well. In addition, from Fig. 6(d) and Fig. 6(e), we observe that i-th branch is prone to predict these parcels whose PLs are i at parcels' center locations. It's because we label the central regions of parcels as positive samples when training the network.

Comparison with Previous Crowded Detection Works. To our knowledge, there is no instance segmentation work in object overlapping scenes. But some detection works in crowded scene have been proposed which are similar with our work. Therefore, in this experiment, we give a comparison with two object detection works—Soft-NMS [1] and CrowdDet [3]—using object detection metrics. CrowdDet is the state-of-the-art method in crowded detection. We re-implement the Soft-NMS according to original paper [1]. For CrowdDet, we use the official open-source implementation[3]. Both methods use the initial learning rate 0.01. Other settings are the same as our approach.

Table 4 shows the comparison results. For our instance segmentation method, we can obtain predicted bounding boxes through outputs of box branch or predicted masks, so we report our results based on both ways (the last row and the second to the last row in Table 4, respectively).

According to Table 4, our method improves about 1–2% in AP, 1–3% in $recall$, and 5–8% in MR^{-2} compared to CrowdDet and Soft-NMS, which further indicates our approach is very effective to deal with these scenes again.

Table 4. Comparison with previous crowded detection works

Method	AP	$Recall$	MR^{-2}
FPN + Soft NMS [1]	94.88	97.67	20.55
CrowdDet (with RM) [3]	94.22	96.00	17.85
Ours (predicted masks)	95.62	**98.04**	13.85
Ours (box branch)	**95.71**	97.96	**12.68**

4.4 Ablation Studies

In this section, we study the impact of different shared layer numbers in multi-level heads.

As mentioned in Sect. 3.2, in each multi-level head, we share the first few layers between branches to make network parameter-efficient. It's easy to see that the shared layers try to capture the common features among different PLs and the latter layers try to capture specialized features only used in a single PL. Table 5 shows the results. The AP and MR^{-2} improves slowly with the decrease of shared layer number, but $recall$ decreases slightly. After shared layer number reduces to one, the gains become much small, and thus setting the number of shared layers to one is a nice choice.

[3] https://github.com/Purkialo/CrowdDet.

Table 5. Results with different number of shared layers in multi-level heads

Number	AP	AP_{OL}	$Recall$	$Recall_{OL}$	MR^{-2}
4	96.25	95.06	**98.38**	**98.06**	13.24
3	96.30	95.60	98.04	97.69	11.01
2	96.26	95.56	97.70	97.37	10.33
1	**96.46**	**95.93**	98.01	97.73	**9.68**
0	96.39	95.92	97.83	97.57	9.75

5 Conclusion

In this work, we propose a new instance segmentation framework based on a multi-level prediction mechanism for segmenting overlapped parcels. The multi-level prediction mechanism is implemented by extending prediction heads to multiple branches. In addition, bounding-box is abandoned to improve the segmentation performance of overlapped parcels. Finally, the experimental results have demonstrated the effectiveness of our proposed method.

Theoretically, our approach also can be applied to other box-free instance segmentation frameworks like SOLOv2. This still needs to be done.

References

1. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-NMS-improving object detection with one line of code. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5561–5569 (2017)
2. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: YOLACT: real-time instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9157–9166 (2019)
3. Chu, X., Zheng, A., Zhang, X., Sun, J.: Detection in crowded scenes: one proposal, multiple predictions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12214–12223 (2020)
4. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2011)
5. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2961–2969 (2017)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. He, Y., Zhu, C., Wang, J., Savvides, M., Zhang, X.: Bounding box regression with uncertainty for accurate object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2888–2897 (2019)

9. Hosang, J., Benenson, R., Schiele, B.: A convnet for non-maximum suppression. In: Rosenhahn, B., Andres, B. (eds.) GCPR 2016. LNCS, vol. 9796, pp. 192–204. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45886-1_16
10. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4507–4515 (2017)
11. Huang, X., Ge, Z., Jie, Z., Yoshie, O.: NMS by representative region: Towards crowded pedestrian detection by proposal pairing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10750–10759 (2020)
12. 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, pp. 6409–6418 (2019)
13. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
14. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
15. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
16. Liu, S., Huang, D., Wang, Y.: Adaptive NMS: Refining pedestrian detection in a crowd. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6459–6468 (2019)
17. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
18. Qi, L., Liu, S., Shi, J., Jia, J.: Sequential context encoding for duplicate removal. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 2053–2062 (2018)
19. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
21. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
22. Shao, S., et al.: CrowdHuman: a benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018)
23. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 282–298. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_17
24. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
25. Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L.: SOLO: segmenting objects by locations. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020.

LNCS, vol. 12363, pp. 649–665. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58523-5_38
26. Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., Shen, C.: Repulsion loss: detecting pedestrians in a crowd. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7774–7783 (2018)
27. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: SOLOv2: dynamic and fast instance segmentation. Adv. Neural Inf. Process. Syst. **33**, 17721–17732 (2020)
28. Zhang, S., Benenson, R., Schiele, B.: CityPersons: a diverse dataset for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3221 (2017)
29. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Occlusion-aware R-CNN: detecting pedestrians in a crowd. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 637–653 (2018)