Mask Point R-CNN

Wenchao Zhang Chong Fu and Mai Zhu

***Abstract*—The attributes of object contours has great significance for instance segmentation task. However, most of the current popular deep neural networks do not pay much attention to the target edge information. Inspired by the human annotation process when making instance segmentation datasets, in this paper, we propose Mask Point R-CNN aiming at promoting the neural networks attention to the target edge information, which can heighten the information propagates between multiple tasks by using different attributes features. Specifically, we present an auxiliary task to Mask R-CNN, including utilizing keypoint detection technology to construct the target edge contour, and enhancing the sensitivity of the network to the object edge through multi-task learning and feature fusion. These improvements are easy to implement and have a small amount of additional computing overhead. By extensive evaluations on the Cityscapes dataset, the results show that our approach outperforms vanilla Mask R-CNN by 5.4% on the validation subset and 5.0% on the test subset.**

***Index Terms*—Instance segmentation, multi-task learning, neural networks, objects contour.**

I. Introduction

In recent years, due to the emergence of deep networks, the performance of many computer vision applications has been dramatically improved. Classification, object detection, and instances segmentation are hot topics in computer vision, especially instances segmentation, which includes two tasks: instance localization and segmentation. Compared with target detection and image level classification, it requires more sophisticated annotation information, therefore, its framework is more complicated.

At present, like Mask R-CNN [1] and MaskLab [2], the champion algorithms of COCO dataset [3] segmentation task, the main method of instance segmentation is to classify the objects in the bounding boxes at the pixel level. In this paper, we focus on improving the segmentation task of Mask R-CNN, which is based on Faster R-CNN [4], after samples the region of interest obtained by the feature extractor through RoIAlign as the feature map with the spatial scale of 14×14, a mask head is added to train the network through a series of convolution and up-sampling operations to finally obtain a prediction mask with the spatial scale of 28×28. But this kind of prediction result is obtained through binary classification and thereby cannot well represent the geometric properties of the target. To address this drawback, we made some improvements on Mask R-CNN. In the framework design of Mask R-CNN, it can not only perform target segmentation, but also perform keypoint detection. Our method is very simple, the target edge points were predicted by using its keypoint detection pipeline as an auxiliary task. Specifically, this work presents a novel method of achieving instance segmentation by fusing target edge information. In detail, we use mask head pixel-by-pixel classification as the main task and focus on the target edge information by aggregating the target edge points simutanously. This multi-task joint training model can well transform the geometric features of the target into the constraints of the binary mask of instance segmentation, further sharpen the edges of the target segmentation results, and improve the detection performance of the model.

In summary, the main contributions of this work are highlighted as follows:

1. We propose the Mask Point R-CNN, illustrated in Fig. 1, which is a new exploration of instance segmentation by combining the information about the aggregation point of the target edge.

2. Compared to Mask R-CNN, we utilize features in the mask prediction branch and keypoint prediction branch with different attributes, and perform joint training through multi-tasks features fusion.

The rest of this paper is organized as follows. In Section II, we briefly review related work on instance segmentation, multi-task learning, and keypoint estimation. In Section III, we describe the motivation and details of our proposed algorithm. Experimental details and analysis of the results are elaborated in Section IV. In Section V, we analyze the interaction between our method subtasks and verified their effectiveness through visualization. Finally, we conclude the paper in Section VI.

II. Related Work

This section provides an overview of literatures on instance segmentation, multi-task learning, and keypoint estimation.

A. Instance segmentation

There are two common ways of achieving instance segmentation: detection first and segmentation first. The detection first method consists of using the target detector to obtain the bounding boxes coordinates preferentially and then segment the instances in the bounding boxes. He *et al*. [1] proposed Mask R-CNN to perform the segmentation task by adding a mask prediction branch on the basis of object classification and bounding boxes regression of Faster R-CNN [4]. Many subsequent methods were proposed based on Mask R-CNN. For instance, Chen *et al*. [2]proposed MaskLab that utilized semantic segmentation and direction prediction to implement mask prediction. Huang *et al*. [5] proposed MS R-CNN(Mask Scoring R-CNN) that added a mask IoU(Intersection over Union) head by combining instance features and corresponding prediction masks in Mask R-CNN to enhance the consistency between mask quality and mask score. Liu *et al*. [6]

W. Zhang, C. Fu and M. Zhu are with the School of Computer Science and Engineering, Northeastern University, China 110819 (e-mail: 1910597@stu.neu.edu.cn; fuchong@mail.neu.edu.cn; zhumai@stumail.neu.edu.cn).

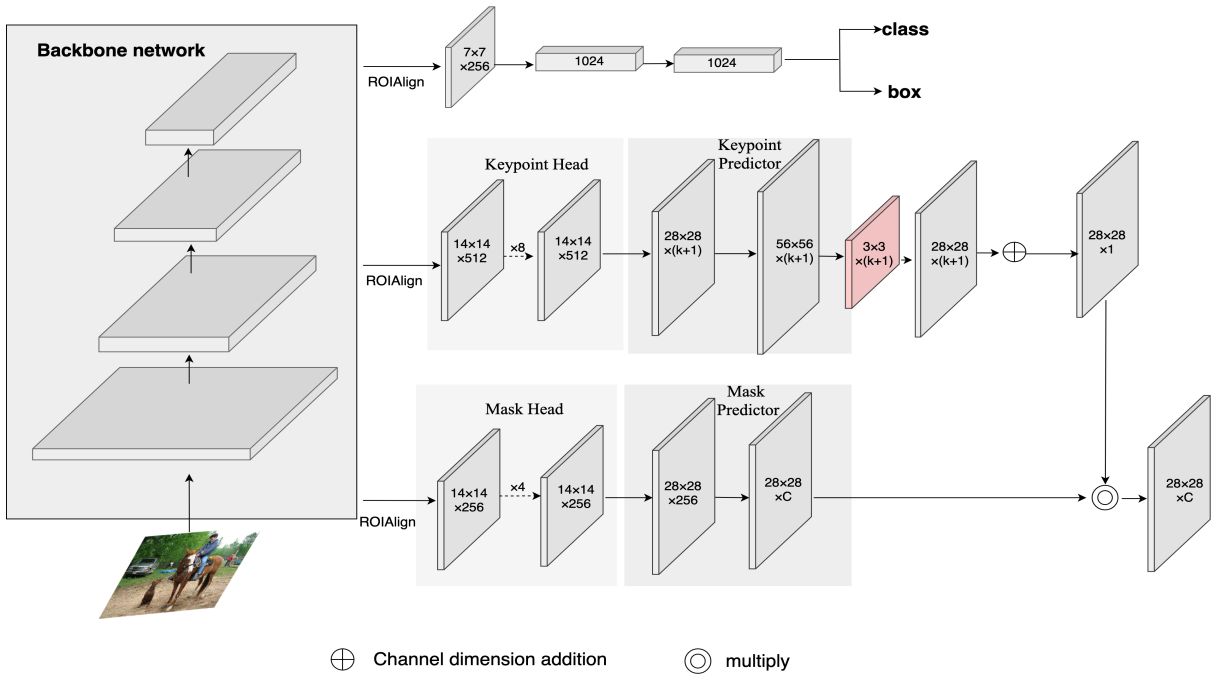

Fig. 1. Network architecture of Mask Point R-CNN. The backbone network is in charge of image processing and feature extraction generate RoIs through RPN which was subsequently corrected to a fixed spatial scale by RoIAlign.

proposed PAN(Path Aggregation Network) which combined more low-level features through Bottom-up Path Augmentation, and obtained more accurate segmentation through Adaptive Feature Pooling and Fully-connected Fusion. Contrary to the previous method, the segmentation first method is to classify each pixel in the image preferentially and then group them into a single instance. For instance, Hariharan *et al.* [7] proposed SDS(Simultaneous detection and segmentation) to locate segmentation candidate regions. This method uses multi-scale combinatorial grouping [8] and obtain bounding boxes simultaneously to achieve instance segmentation. The InstanceCut [9] method utilized the edge graph to divide the segmentation map into different objects. Our work is based on Mask R-CNN and have improved it in several ways.

B. Multi-Task learning

MTL (Multi-Task Learning) is an inductive transfer method that makes full use of domain-specific information hidden in the training signals of multiple related tasks. During backward propagation, multi-task learning allows shared hidden layer-specific features to be used by other tasks. MTL will learn features that are applicable to several different tasks. Such features are often not easy to learn in a single-task learning network. MTL is widely used in computer vision [10]–[13], speech, natural language processing, and other fields [14], [15]. In [10], Su *et al.* calibrates missing features in the learning process by combining low-level raw binary attributes and intermediate attributes. In [11], Abdulnabi *et al.* proposed the use of a combination of hidden matrix and multi-matrix to decouple model parameters to allow CNN (Convolutional Neural Network) models to share visual knowledge between different attribute categories Simultaneously. In [12], Yuan *et al.* enhanced the robustness of model coefficient estimation by constructing a hierarchical sparse learning model of multi-task interconnection. In [13], Yim *et al.* added an additional task to improve the ability of DNN (Deep Neural Network) to maintain face identity. Taking a human face and a binary code representing the target pose as input to generate a face image with the same identity and target pose. In our paper, we train the Mask R-CNN output binary classification information of the target and combine the edge keypoint information of the training target to calibrate the instance segmentation results.

C. Keypoint estimation

Keypoint estimation technology has been widely used in computer vision research, such as human pose detection tasks [16]–[19] and anchor free target detection tasks [20]–[23]. Human pose estimation mainly detects k 2D keypoints of the human body, such as joints, facial features, etc., and describes human bone information by these keypoints. As mentioned above, keypoint estimation techniques are used in many detection tasks. For a detection first instance segmentation method, it needs to obtain the target bounding box coordinates through the detection network preferentially, the current target detection method can be divided into anchor base [4], [24]–[28] and anchor free according to whether the region proposal is obtained by sliding window. The anchor base method utilizes sliding window to generate a bunch of candidate anchor boxes, and then remove the redundant bounding boxes by NMS (Non-Maximum Suppression) [29] to get the final detection results. The anchor free method utilizes keypoint detection technology [30], [31] to obtain the center point or extreme point of the target for target positioning. However, the detection points of this method are all located

Fig. 2. Instance segmentation results of our model on the Cityscapes validation set, top row: input images; middle row: keypoint detection results; bottom row: joint training detection results.

in the interior of the detected object, so it is more conducive to the learning of network features. Inspired by this method, we utilize the keypoint branch of Mask R-CNN to detect the edge aggregation points of the target to be segmented.

III. Method

In this section, we will first introduce the motivation of our design, then briefly review the Mask R-CNN algorithm, and finally describe the structure of our model in detail from three aspects: boundary point extraction, model framework, and loss function.

A. Motivation

Since most of the existing instance segmentation methods are implemented by pixel-by-pixel classification, these methods depend on finer label information relative to the target detection task. Thus, making a target segmentation dataset will be a major expenditure of time and effort. During the annotation process, the annotator needs to obtain the edge contour of the instance by connecting the edge points for building the instance's mask, so the edge information of the instance can well represent the instance. In order to improve the accuracy of instance segmentation when the amount of data is limited, inspired by the artificial annotating process, we obtain a fixed number of edge aggregation points by uniformly sampling the edge contours of the groundtruth mask, and then utilize the keypoint prediction branch of Mask R-CNN to predict these edge aggregation points. Finally, the binary segmentation mask is obtained by fusing the edge information prediction and the mask prediction.

B. Mask R-CNN

Mask R-CNN expands on the basis of Faster R-CNN, and in parallel adds a new branch for predicting the object mask on the bounding box recognition branch. The network can also be easily extended to other tasks, such as estimating the pose of a person, that is, person keypoint detection. The structure of its mask head and keypoint head is similar. The former obtains an output resolution of 28×28 through four 3×3 convolutional layers with a dimension of 256 followed by an up-sampling layer, and finally through a convolutional layer to make its dimensions consistent with the target category. Meanwhile, the latter obtains an output resolution of 56×56 through eight 3×3 convolutional layers with 512 dimensions followed by two upsampling layers. In order to implement Human Pose Estimation in Mask R-CNN, it uses one-hot mask to encode each keypoint position, and the structural design is similar to the mask branch, which uses fully convlution network to predict a binary mask for each keypoint.

C. Boundary point extraction

To train our keypoint detection subtask, we need the labels of the target edge points. But the dataset used for instance segmentation only provides binary groundtruths masks, so we need to make labels for keypoint detection subtasks ourselves. We can get the contour of the target from the binary mask of the target according to [32]. In the keypoint detection task, a heat map is generated for each detection point. For different instance targets in the training data, we need to extract the contour boundary points of the target to form training labels. However, the CNN network reqiure the input training data with a constant channel dimension, so we also need to extract a fixed number of boundary points. We generate training labels for keypoint detection by sampling the instance target contours, as shown in the Fig. 3. But due to the large difference in the size of the instance targets in the training dataset, the number of points obtained by sampling may be small. In this regard, we complement by randomly selecting existing sampling points. We also explored the different sampling method, which will be explained in detail in the experimental section.

TABLE I
RESULTS ON CITYSCAPES VAL SUBSET, DENOTED AS AP [VAL], AND ON CITYSCAPES TEST SUBSET, DENOTED AS AP

Methods	AP[val]	AP	AP_{50}	person	rider	car	truck	bus	train	motorcycle	bicycle
Kendall *et al.* [33]	-	21.6	39.0	19.2	21.4	36.6	18.8	26.8	15.9	19.4	14.5
Arnab *et al.* [34]	-	23.4	45.2	21.0	18.4	31.7	22.8	31.1	31.0	19.6	11.7
SGN [35]	-	25.0	44.9	21.7	20.0	39.4	24.7	33.2	30.8	17.7	12.3
PolygonRNN++ [36]	-	25.4	45.4	29.3	21.7	48.2	21.1	32.3	23.7	13.6	13.6
Neven *et al.* [37]	-	27.6	50.8	34.5	26.0	52.4	21.6	31.1	16.3	20.0	18.8
GMIS [38]	-	27.6	44.6	29.2	24.0	42.7	25.3	37.2	32.9	17.5	11.8
BshapeNet+ [fine-only] [39]	-	27.3	50.4	29.7	23.3	46.7	26.0	33.3	24.8	20.3	14.1
Mask R-CNN[fine-only] [1]	31.5	26.2	49.9	30.5	22.7	46.9	22.8	32.2	18.6	19.1	16.0
Ours[fine-only]	36.9	31.2	56.1	37.3	28.2	55.9	28.6	35.1	23.7	21.3	19.8

D. Model framework

According to the definition in [40], we can use symbols $\{\tau_i\}_{i=1}^{m}$ to describe multi-task training deep learning models, where m represents the number of subtasks. For our framework $m = 4$, it includes the bounding box regression task, the classification task, the target segmentation task, and the new keypoint detection task. Suppose that our dataset D_i contains a total of n_i training samples, then $D_i = \{x_i^j, y_i^j\}$, where $x_i^j \in \mathbb{R}^{d_i}$ is the $j-th$ training instance in τ_i and y_i is its corresponding training label. In our approach, we used the same training data for different learning tasks with different training labels, i.e. $x^i = x^j, y^i \neq y^j, and\ i \neq j$. More formally, the training label for the regression task of the bounding box is the coordinate of the center point of the groundtruth and its width and height. And for the target segmentation task, the training label is the corresponding binary mask, whereas for the keypoint detection task, the training label is the edge aggregation point obtained by sampling the target boundary.

Fig. 1 shows the general architecture of Mask Point R-CNN. The faster r-cnn head is standard component of Mask R-CNN. We can simultaneously perform keypoint joint training with faster r-cnn head and mask prediction branch of Mask R-CNN. Suppose that we utilize k edge aggregation points and the center point of the target, a total of $k+1$ points, as the keypoint labels information for training. And a $M \times M$ heat map is generated for each keypoint in the keypoint detection task of Mask R-CNN, so we can get a $M \times M \times (k+1)$ output mask tensor eventually. At the mask prediction branch, a $N \times N \times C$ output tensor is produced through the fully convolutional network, where C represents the total number of object categories. In addition, to capture the overall edge information of the target, we do the channel-wise addition on the output features of the keypoint prediction branch for mapping the information of multiple keypoint with one hot encoding onto a single heat map. And in the next steps we reduce the heat map into a feature map with the same spatial scale as the mask prediction branch output by using a 3×3 convolution layer. In the end we broadcaste the edge information of the target to each channel of the mask prediction branch through feature fusion. There is one more thing should notice that we do not use ReLU activation function after the convlution layer uesd for reducing feature map spatial scale.

The above fusion process could be mathematically described as follows:

$$O_k = \sum_{i=1}^{i=k+1} Conv_{3\times3}^{s=2}(f_{k+1}), \tag{1}$$

$$O_m = O_k \otimes f_m, \tag{2}$$

where $\otimes$ represents channel-wise multiplication, s stands for the stride of the convolution operation, f_{k+1} stands for the output features of the keypoint prediction branch, and f_m stands for the output features of the mask prediction branch.

E. Loss function

In Mask R-CNN, the multi-task loss is defined as three parts: L_{cls}, L_{box}, and L_{mask}. For our model we calculate the loss in the keypoint detection task by using the cross entropy function, which is the same as defined in [1]. After the keypoint detection is added to our model, the loss function can be updated as follows:

$$L = L_{cls} + L_{box} + L_{mask} + \alpha L_{keypoint}, \tag{3}$$

where α represents the weight parameter to balance the loss of the keypoint detection task.

(a) (b) (c)

Fig. 3. Examples of target contour point sampling. (a) Target binary mask label. (b) Corner sampling. (c) Uniform sampling.

IV. EXPERIMENTS

In this section, we will first introduce our experimental dataset, evaluation criteria, and model parameter settings. Then we conducte ablation experiments from two aspects, namely the edge feature fusion and influence of keypoint. Moreover, we report the results of instance segmentation only using the edge features of the keypoint.

A. Dataset and metrics

We verify our algorithm on the Cityscapes dataset, which contains a variety of stereo video sequences recorded in street scenes from 50 different cities. It provides 5000 images with fine annotations and a fixed resolution of 2048×1024, which are splited into 2975, 500 and 1525 images for train/val/test, and the test dataset labels were not publicly released. We evaluate results based on AP (Average precision) and AP_{50}. We did plenty of ablation experiments on the validation set and provided results on the test dataset. We compare our model with other state-of-the-art methods on Cityscapes test subset, and the results are shown Table I. Training on fine-only data, our method outperforms Mask R-CNN by 5.0% on test subset.

B. Implementation Details

We perform our experiments using the implementation of Mask R-CNN based on the torchvision detection module with a pytorch backend [41]. We take 4 images in one image batch for training. The shorter edges of the images are randomly sampled from [800, 1024] for reducing overfitting. Each training is carried out on two nvidia titan xp GPUs. For instance segmentation task the mask spatial scale is 28×28, and for keypoint detection task is 56×56. We randomly horizontal flip the input image with a probability of 0.5 to augment the dataset. Because we use the different number of GPUs, different training strategies are used for Citysacpes dataset compared with Mask R-CNN. We train our model a total of 64 epochs, with a learning rate of 0.005 for the fore 48 epochs and 0.0005 for the remaining 16 epochs. Following Mask R-CNN, we use SGD for gradient optimization with weight decay 0.0001 and momentum 0.9. ResNet-50 FPN [42] is taken as the initial model on this dataset, if not specially noted. And for keypoint detection task we use uniform sampling with 100 sampling points. In order to be able to make a fair comparison, we do not wait for the loss to completely converge for all the experiments, but train our model according to the training strategy described above.

C. Edge Feature Fusion

The structural design choices for feature fusion: In our framework, feature fusion between multiple tasks is required, but the mask spatial scale of instance segmentation task and keypoint detection task is different, 56×56 vs 28×28. Therefor, before the fusion, we need to ensure that the spatial scales of the two subtasks are consistent. There are a few design choices shown in Fig. 4 and explained as follows:

(a) The output of keypoint predictor is size of $56 \times 56 \times (k+1)$, and the output of mask predictor is size of $28 \times 28 \times C$. After we do channel-wise addition on the keypoint predictor output, the feature map spatial scale is reduced to 28×28 using a 3×3 convolution layer with a stride of 2.

(b) The output of keypoint predictor is size of $56 \times 56 \times (k+1)$, and the output of mask predictor is size of $28 \times 28 \times C$. Before we do channel-wise addition on the keypoint predictor output, the feature map spatial scale is reduced to 28×28 using a 3×3 convolution layer with a stride of 2.

(c) The output of keypoint predictor is size of $56 \times 56 \times (k+1)$, and the output of mask predictor is size of $28 \times 28 \times C$. we upsample the output spatial scale of the mask predictor to 56×56 using deconvolution.

(d) The output of keypoint predictor is size of $28 \times 28 \times (k+1)$, and the output of mask predictor is size of $28 \times 28 \times C$.

TABLE II
THE STRUCTURAL DESIGNS RESULTS ON CITYSCAPES VAL SUBSET

Setting	AP	AP_{50}
Mask R-CNN re-implement	33.1	60.0
structure design (a)	31.8	59.5
structure design (b)	36.0	63.0
structure design (c)	35.6	62.3
structure design (d)	34.9	62.1

TABLE III
THE SPATIAL SCALE REDUCTION DESIGES RESULTS ON CITYSCAPES VAL SUBSET

Setting	AP	AP_{50}
Mask R-CNN re-implement	33.1	60.0
maxpooling	34.1	61.5
avgpooling	33.6	60.7
3x3 convlution	36.0	63.0

The results are shown in Table II. We can see that each methods can achieve performance improvement except method (a), because in method (a) we only use one convolution layer to process the prediction results with a single channel, losing a lot of original edge prediction information.

The spatial scale reduction design choices for feature fusion: Based on the design of the edge fusion structure, we try to reduce the spatial scale of the keypoint prediction output by using three methods: maxpooling, avgpooling, and a convolution layer with kernel size of 3 and stride of 2.

Table III shows the results for above spatial scale reduction designs, we can see that all the three methods can improve the performance of the model, further validating the effectiveness of our method. At the same time, we can see that it can better integrate edge prediction information than simple pooling by using the method of convolution to learn parameters for reducing the feature spatial scale.

The choices of the fusion mode: After channel-wise addition and spatial scale reduction of the keypoint branch output, we can perform the fusion of edge features. We test three methods of maximum, multiplication, and addition to choose the best fusion strategy. Results are shown in Table IV, we can see that it can maximize the effect of the model by using multiplication.

D. Influence of keypoint

The model we designed requires training tasks including faster r-cnn, mask head, and keypoint head. But the Cityscapes

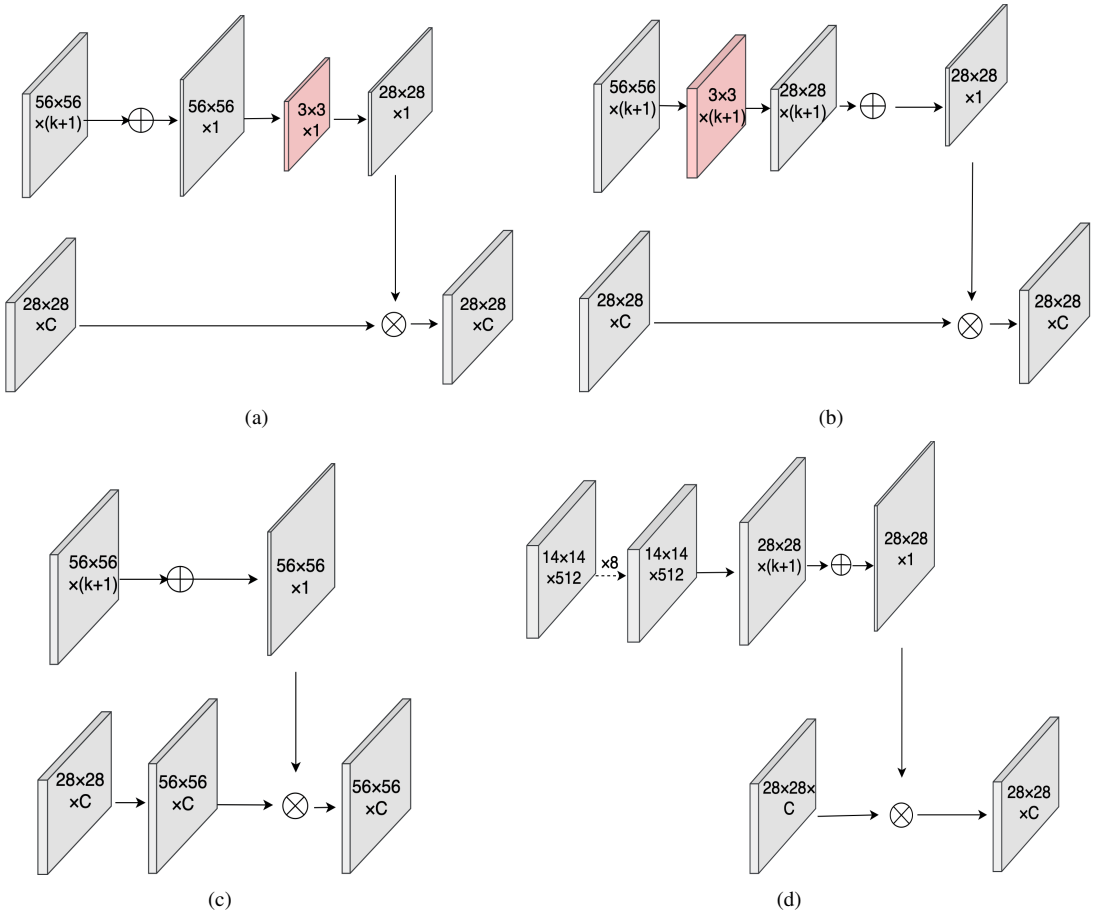

Fig. 4. Different structural design choices for feature fusion.

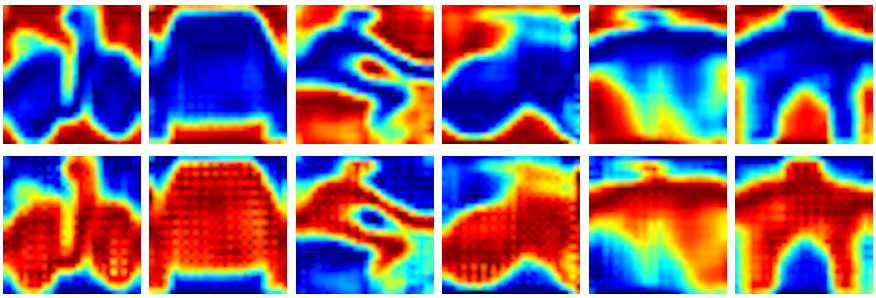

Fig. 5. Examples of instance heat map, top row: keypoint branch output feature maps; bottom row: mask branch output feature maps.

TABLE IV
THE SPATIAL SCALE REDUCTION DESIGES RESULTS ON CITYSCAPES VAL SUBSET

Setting	AP	AP_{50}
Mask R-CNN re-implement	33.1	60.0
add	34.3	61.5
max	31.9	61.0
multiply	36.0	63.0

TABLE V
THE DIFFERENT SAMPLING POINTS RESULTS ON CITYSCAPES VAL SUBSET

Number of samples	AP	AP_{50}
Mask R-CNN re-implement	33.1	60.0
k = 50	34.5	61.7
k = 100	36.0	63.0

dataset only provides instance groundtruths masks, so we need to get the bounding box and edge aggregation point labels of the target from the mask of the instance ourselves. For a single target, the bounding box label is uniquely fixed, but for the edge aggregation point label we need to sample it from the edge contour of the target. We separately analyze the impact of corner sampling and uniform sampling on our method, and the results are shown Table VI. Generally speaking, corner sampling reflects the key geometric information of the object, whereas uniform sampling reflects the overall geometric in-

TABLE VI
THE DIFFERENT SAMPLING TYPES RESULTS ON CITYSCAPES VAL SUBSET

Type of samples	AP	AP_{50}
Mask R-CNN re-implement	33.1	60.0
corner	35.1	62.1
uniform	36.0	63.0

TABLE VII
MULTI-TASK LEARNING OF BOX, MASK, AND KEYPOINT, EVALUATED ON CITYSCAPES VAL SUBSET USING COCO EVALUATION METRICS.

Methods	AP_{bb}	AP_{mask}	$AP_{keypoint}$
Mask R-CNN, mask-only	37.7	32.4	-
Mask R-CNN, keypoint-only	36.8	-	55.7
Ours (without center point)	36.7	34.3	56.0
Ours	37.2	35.2	56.4
Ours ($\alpha = 0.5$)	37.9	35.9	58.2

formation of the object. We can see from the experimental results that it is more conducive to the instance segmentation by using keypoint detection technology to capture the overall geometric position information of the object. If the number of edge sampling points is less than the required number due to the target being too small when sampling the edge points, we randomly select the existing sampling points for padding.

Furthermore, we demonstrate the influence of different sampling points number on the model's performance, the results are shown in Table V. Specifically, we conducted experiments on 50 and 100 sampling points, respectively. And we find in the course of the experiment that the further increase of the sampling points number will damage the effectiveness of the model, which can be attributed to we add too much random information in padding operation, since there are a large number of small targets in the dataset. In our approach, we use not only the points sampled from the target contour but also the center point of the target when we performe the keypoint estimation task, this is because we hope that the geometric relationship between the target boundary point and the center point can be learned during the model training process, thereby improving the estimation accuracy of the boundary point. For example, the center point of the target can be regarded as the origin in a polar coordinate system, for the targets with the same category, the distance and angle of the target boundary point relative to the center point have certain regularity. To verify our analysis, we verify the effect of adding the target center point using coco evaluation metrics, as shown in the Table VII, where AP_{bb} represents the AP value of the bounding box detection branch, AP_{mask} and $AP_{keypoint}$ represent the AP value of the mask branch and keypoint estimation branch respectively. we can see that the performance of all three subtasks has been improved after adding the center point.

E. Other experiment

In the joint training, we can get the edge point information of the target through the keypoint estimation task. Actually, the target can be segmented by only using the contour information mapped by these edge points. When $k = 100$, we can get AP value of 12.5 on Cityscapes val subset.

V. DISCUSSION

In this section, we will analyze the effectiveness of our method from the interaction of multi-task joint training.

In fact, in Mask R-CNN, the author has also carried out multi-task joint training of target detection, keypoint estimation, and segmentation on the person category in the COCO dataset. However, the experimental results show that only the performance of the keypoint estimation task is slightly improved, whereas both the performance of the detection and segmentation tasks is degraded. This is because in the task of human pose estimation, the labels of keypoints are located within the instances of the human body, and the overall information of the instances captured by the segmentation task can help locate these keypoints. But for the target detection task including positioning and classification, and the segmentation task composed of pixel-wise classification, the keypoint information does not help, But after we set the weight balance parameters of the keypoint detection task $\alpha = 0.5$, we can see that the detection performance of these three tasks has been greatly improved.

However, the keypoint constructed in our experiment is tightly attached to the contour edge of the instance, so during the training process, the performance of the segmentation task can be improved by enhancing the model's attention to the target's edge. We remeasure the results of our model on the Cityscapes val subset using COCO evaluation metrics in order to verify our analysis, the results are shown in Table VII, we can see that our method can simultaneously improve the performance of instance segmentation, keypoint estimation and object detection tasks.

Moreover, in order to verify the effectiveness of our method more intuitively, we visualize the final feature map obtained by the keypoint branch and the corresponding class feature map obtained by the mask branch. Specifically, we normalize the two feature maps respectively and generate corresponding heat maps, some examples are demonstrated in Fig. 5(the darker pixels in the heat map represent higher scores). From these examples we can infer that the keypoint output feature map obtained by channel-wise addition can not only learn the boundary information of the target, but also get the overall background information of the target. Therefore, the final output feature of the mask branch obtained by fusing the two tasks is equivalent to segmenting the target by combining the background and foreground.

VI. CONCLUSION

In this paper, we propose Mask Point R-CNN for instance segmentation, which can enhance the consistency between segmentation prediction results and groundtruths masks by adding edge aggregation point auxiliary detection tasks. The ablation experiments on the Cityscapes dataset have proved the effectiveness of our method. We hope that our method can have some inspiration for other instance segmentation methods.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[2] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "Masklab: Instance segmentation by refining object detection with semantic and direction features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4013–4022.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[5] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6409–6418.

[6] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[7] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 297–312.

[8] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 328–335.

[9] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multicut," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5008–5017.

[10] C. Su, F. Yang, S. Zhang, Q. Tian, L. S. Davis, and W. Gao, "Multi-task learning with low rank attribute embedding for person re-identification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3739–3747.

[11] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task cnn model for attribute prediction," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1949–1959, 2015.

[12] C. Yuan, W. Hu, G. Tian, S. Yang, and H. Wang, "Multi-task sparse learning with beta process prior for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 423–429.

[13] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim, "Rotating your face using multi-task deep neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 676–684.

[14] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4460–4464.

[15] L. Zhao, Q. Sun, J. Ye, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Feature constrained multi-task learning models for spatiotemporal event forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 1059–1072, 2017.

[16] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2334–2343.

[17] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4929–4937.

[18] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 466–481.

[19] D. Osokin, "Real-time 2d multi-person pose estimation on cpu: lightweight openpose," *arXiv preprint arXiv:1811.12004*, 2018.

[20] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[21] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 850–859.

[22] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[23] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 616–625.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[25] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[27] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[28] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[29] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5561–5569.

[30] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.

[31] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.

[32] W. Lorensen and H. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, pp. 163–, 08 1987.

[33] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[34] A. Arnab and P. H. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 441–450.

[35] S. Liu, J. Jia, S. Fidler, and R. Urtasun, "Sgn: Sequential grouping networks for instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3496–3504.

[36] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

[37] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8837–8845.

[38] Y. Liu, S. Yang, B. Li, W. Zhou, J. Xu, H. Li, and Y. Lu, "Affinity derivation and graph merge for instance segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 686–703.

[39] B. R. Kang, H. Lee, K. Park, H. Ryu, and H. Y. Kim, "Bshapenet: Object detection and instance segmentation with bounding shape masks," *Pattern Recognition Letters*, vol. 131, pp. 449–455, 2020.

[40] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.