Joint object contour points and semantics for instance segmentation

Wenchao Zhang¹ | Chong Fu¹,²,³ ⚫ | Mai Zhu¹ | Lin Cao⁴ | Ming Tie⁵ | Chiu-Wing Sham⁶

¹School of Computer Science and Engineering, Northeastern University, Shenyang, China
²Engineering Research Center of Security Technology of Complex Network System, Ministry of Education, Shenyang, China
³Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, Northeastern University, Shenyang, China
⁴School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing, China
⁵Science and Technology on Space Physics Laboratory, Beijing, China
⁶School of Computer Science, The University of Auckland, Auckland, New Zealand

Abstract
The edges of objects are of great significance to the task of instance segmentation. However, most of the current popular deep neural networks do not pay much attention to the object edge information. More importantly, using the down-sampling pooling layer in the deep learning network, the edge detail information of the object will be lost. To address this issue, inspired by the manual annotation process, we propose Mask Point R-CNN aiming at promoting the neural network’s attention to the object boundary. Specifically, we introduce the auxiliary task of object contour point detection on the Mask R-CNN framework, which can effectively improve the gradient flow between different tasks by multi-task learning and repairing objects' boundary information via feature fusion. Consequently, the model can be more sensitive to the edges of the object and capture more geometric features. Quantitatively, the experimental results show that our Mask Point R-CNN outperforms vanilla Mask R-CNN by 3.8% on the Cityscapes dataset and 0.8% on the COCO dataset.

Keywords
feature fusion, instance segmentation, multi-task learning, object boundary

1 | INTRODUCTION

In recent years, due to the emergence of deep networks, the performance of many computer vision applications has been dramatically improved (Khalifa et al., 2021; Patel & Vidyarthi, 2021; Phong et al., 2021; Samanta et al., 2021). Classification, object detection, and instances segmentation are hot topics in the computer vision community, especially instances segmentation, which includes two tasks: instance recognition and semantic segmentation. Compared with object detection and image-level classification, it requires more sophisticated annotation information, and thus its framework needs a more elaborate design.

Mask R-CNN (He et al., 2020), as an outstanding representative of the instance segmentation framework, its main idea is to divide the instance segmentation pipeline into two stages: the first stage is to locate and classify the objects, and the second stage is to classify the foreground and background pixel by pixel. Mask R-CNN is a variant of Faster R-CNN (Ren et al., 2015) algorithm, so it inherits the advantages of thinking twice for generating a more compact bounding box. Unfortunately, for instance segmentation tasks, it ignores the attention to the boundaries of objects that are difficult to segment. Specifically, Mask R-CNN adds a parallel mask head to the original Faster R-CNN box head. In a nutshell, like Faster R-CNN, it first obtains the RoIs (region of interest) with a spatial scale of 14 × 14 through RoIAlign operation. Then a mask tensor with a spatial scale of 28 × 28 is predicted by using a series of convolution and upsampling operations. But this kind of prediction result is
only obtained through binary classification and thereby cannot sufficiently represent the geometric properties of the object. Therefore, it is difficult to repair the loss of instance boundary details caused by the down-sampling pooling layer. To address this drawback, we added an auxiliary task based on Mask R-CNN to preserve these boundary details. Similarly, BFP (Ding et al., 2019) suggests utilizing boundary information as a control condition to facilitate scene segmentation. The proposed boundary-aware feature propagation module enhances the similarity of intra-class features while preserving the dissimilarity of inter-class features. BFP treats object boundaries as new semantic categories, which generate comprehensive boundary information for the entire image. However, for the instance segmentation task, it is necessary to differentiate individual objects. Therefore, we adopt the approach of extracting key points from instance objects to generate boundary contours.

As is known, for the person category in the COCO dataset, Mask R-CNN can not only perform object detection and instance segmentation. Alternatively, it can also perform human keypoint detection tasks simultaneously (i.e., keypoint R-CNN) using additional human joint labels. Inspired by the process of labelling instance segmentation objects (i.e., the polygonal dividing line of the object is connected by the boundary points of the object), we extend the keypoint detection task to the object boundary points detection. In other words, we add a new keypoint detection auxiliary task to Mask R-CNN, which can enhance the model’s attention to the object boundary by detecting the object’s contour. Our method uses mask head pixel by pixel classification as the main task and further focuses on the object edge information by aggregating the target edge points simultaneously.

Consequently, this multi-task joint trained model can well transform the geometric features of the object into the constraints of the binary mask of instance segmentation and further sharpen the edges of the instance segmentation results to improve the detection performance. Without losing generality, we validate our method on the Cityscapes dataset (Cordts et al., 2016) and the large-scale object detection COCO dataset (Lin et al., 2014). Besides, compared with the Mask R-CNN baseline with ResNet-50, our method can effectively improve the model’s performance. The AP (average precision) value on the Cityscapes dataset is 36.9% (ours) versus 33.1%, and the result on the COCO dataset is 35.4% (ours) versus 34.6%.

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

1. We propose the Mask Point R-CNN, illustrated in Figure 1, which is a new exploration of instance segmentation by combining the edge information of the object.
2. Compared to Mask R-CNN, we use two different features, mask and boundary point, for instance segmentation, and perform joint training via multi-tasks features fusion.
3. Different from the human body keypoint detection, we extend the keypoint detection task to any category of boundary point detection and experimentally prove that it can enhance the network’s attention to the model boundary.

**FIGURE 1** The network architecture of Mask Point R-CNN. In our model, the backbone network is in charge of feature extraction, and we add a new keypoint head to the box head and mask head of Mask R-CNN, which is used to detect the boundary points of objects.
The rest of this article is organized as follows. In Section 2, we briefly review related work on instance segmentation and keypoint detection. In Section 3, we describe the motivation and details of our proposed algorithm. Experimental information and analysis of the results are elaborated in Section 4. In Section 5, we discuss the interaction between our method subtasks and verify their effectiveness through visualization. Finally, we conclude the article in Section 6.

2 | RELATED WORK

This section provides an overview of literatures on instance segmentation and keypoint detection.

2.1 | Instance segmentation

There are two common ways of achieving instance segmentation: detection first and segmentation first. The detection first method uses the object detector to preferentially obtain the bounding boxes' coordinates and then segment the instances in the bounding boxes. As an outstanding representative work, He et al. (2020) proposed Mask R-CNN to perform the instance segmentation task by adding a mask prediction branch on the basis of object classification and bounding boxes regression task of Faster R-CNN (Ren et al., 2015). Many subsequent methods were proposed based on Mask R-CNN (Chen, Pang, et al., 2019; Huang et al., 2019; Li et al., 2021; Liu, Qi, et al., 2018; Nie et al., 2020; Qiao et al., 2020; Tian, Yang, et al., 2020). For instance, Chen et al. (2018) proposed MaskLab that utilized semantic segmentation and direction prediction to generate the final binary mask. Huang et al. (2019) 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, Qi, et al. (2018) proposed PAN (path aggregation network), which combined more low-level features through bottom-up path augmentation, and obtained more accurate segmentation through AFP (adaptive feature pooling). In addition, there are also many excellent detection first methods that focus on building real-time instance segmentation systems (Chen et al., 2020; Lee & Park, 2020; Peng et al., 2020; Tian et al., 2021; Xie et al., 2020; Zheng et al., 2021). Contrary to the previous method, the segmentation first method (Chen, Strauch, & Merhof, 2019; Gao et al., 2021; Kong & Fowlkes, 2018; Zhang & Maire, 2020) is to classify each pixel in the image preferentially and then group them into a single instance. For instance, the InstanceCut (Kirillov et al., 2017) method utilized the edge graph to divide the segmentation map into different objects. Zhang and Maire (2020) first grouped the image into multiple regions for contour detection and then merged these regions into a tree-like hierarchical structure for image segmentation. Analogously, the work aforementioned is to perform segmentation tasks by classifying the foreground and background. Our work is to get a better instance segmentation effect by paying attention to the object's boundary.

2.2 | Keypoint detection

Keypoint detection technology has been widely used in computer vision research, such as human pose detection tasks (Geng et al., 2021; Jiang et al., 2021; Zhang, Chen, & Tao, 2021) and anchor free object detection tasks (Liu, Li, et al., 2021; Tang et al., 2021; Zhou, Wang, & Krähenbühl, 2019; Zhou, Zhuo, & Krahenbuhl, 2019). The human body keypoint estimation task is to detect 2D keypoints of the human body, such as joints, facial features and so forth and describe human bone information by these keypoints. Lately, keypoint estimation techniques are also used in many detection tasks. Generally, the current object detection method can be divided into anchor base (Girshick, 2015; Ke et al., 2020; Kong et al., 2020; Ren et al., 2015; Xiao et al., 2021; Zhang, Wan, et al., 2021) and anchor free according to whether the proposal is obtained by a sliding window. Specifically, the anchor base method utilizes a sliding window to generate a bunch of candidate anchor boxes and then remove the redundant bounding boxes by NMS (non-maximum suppression) (Bodla et al., 2017) to get the final detection results. The anchor free method utilizes keypoint detection technology (Dong et al., 2020; Lu et al., 2020; Xu et al., 2021) to obtain the centre point or extreme point of the object for target positioning. By contrast, the detection points of previous methods are all located in the interior of the detected object for meeting their task. Whereas we utilize keypoint detection as an auxiliary task to detect the edge points of each instance.

3 | METHOD

In this section, we first introduce the motivation of our method and then briefly review the head network of Mask R-CNN and Keypoint R-CNN. Finally, we describe the pipeline of our model in detail from three aspects: boundary point extraction, model framework, and loss function.
### 3.1 Motivation

Since most of the existing instance segmentation methods are implemented by pixel-wise classification, these methods need a finer mask label. However, making an instance segmentation dataset is a major expenditure of time and effort. During the labelling process of the instance label, the annotator needs to obtain the edge contour of the instance by connecting the edge points since the boundary point features of the instance can well represent its geometric information. In addition, in the feature extraction process of the convolutional neural network (CNN), as the network depth increases, it will gradually use pooling technology to downsample the input feature tensor, which will lead to the loss of instance boundary details. We all know that deep learning is a typical data-driven method. Therefore, to improve the accuracy of instance segmentation when the amount of data is limited and preserve the outline details of the instance object, we propose to use the contour point detection of the instance object as an auxiliary task to strengthen the network’s attention to the instance boundary.

### 3.2 Mask head and keypoint head

Next, we briefly review the head network of Mask R-CNN and Keypoint R-CNN. Mask R-CNN is extended based on Faster R-CNN. In addition to the shared backbone network and FPN (feature pyramid networks), it adds a new mask head in parallel for predicting the object mask. Similarly, the Mask R-CNN can also be easily extended to other tasks, such as person keypoint detection (i.e., Keypoint R-CNN). In Keypoint R-CNN, its keypoint head is very similar to the mask head of Mask R-CNN, composed of stacked convolutional and upsampling layers. Specifically, the mask head uses four $3 \times 3$ convolutional layers and is followed by an up-sampling layer to obtain an output tensor with a resolution of $28 \times 28$. Then, it employs a convolutional layer to make its channel dimensions consistent with the object category. For example, the COCO dataset has 81 classes (including background categories), so the mask head can finally get an output tensor of $28 \times 28 \times 81$.

In contrast, the keypoint head uses eight $3 \times 3$ convolutional layers followed by two upsampling layers to obtain an output tensor with a resolution of $56 \times 56$. Besides, implementing human pose estimation in Keypoint R-CNN uses a one-hot mask to encode each keypoint position and a fully convolutional network to predict a binary mask for each keypoint. For example, COCO’s pose estimation dataset has 17 keypoints in different positions, so the keypoint head can finally get an output tensor of $56 \times 56 \times 17$.

### 3.3 Boundary point extraction

To train our boundary point detection auxiliary task, we need the labels of the object edge points. But all dataset used, for instance segmentation task, only provides a binary ground truth mask, so we need to extract the contour points of the instance as the supervision label for joint training. Moreover, the CNN network requires the input training data with a constant channel dimension. Correspondingly, we need to extract a fixed number of boundary points. This work generates training labels for keypoint detection by sampling the instance contours.

In particular, we need to consider two factors when making instance contour point labels: (1) What kind of sampling method is used. (2) How many sampling points are needed. For problem 1, we tried two methods: corner sampling and uniform sampling, as shown in Figure 2. And for problem 2, the number of points obtained by sampling may be smaller than the set quantity due to the significant difference in the scale of the instances in the training dataset. In this regard, we randomly select the existing sampling points for padding. Besides, setting the number of contour points is an essential issue because when the number is large, it will increase the computational overhead. Still, if the number is too small, it cannot sufficiently represent the geometric shape of the object. And we made our decisions on these issues through corresponding experimental analysis in Section 4.

### 3.4 Model framework

According to the definition in Zhang and Yang (2017), we can use symbols $\{r_i\}_{i=1}^m$ to describe multi-task training deep learning models, where $m$ represents the number of subtasks. Our framework $m = 4$ includes the bounding box regression task, the classification task, the target segmentation task, and the newly added keypoint detection task. Suppose that our dataset $D$ contains a total of $n_i$ training samples, then $D = \{x'_j, y'_j\}$, where $x'_j$ is the $j$th training instance in $r_i$ and $y'_j$ is its corresponding training label. In our approach, we used the same training data for different learning tasks with different training labels, that is, $x_i = x'_i, y_i \neq y'_i$, and $(i,l) \in \mathbb{R}^m$. More formally, the training label for the regression task of the bounding box is the coordinate of the centre point of the ground truth and its width and height. And for the instance segmentation task, the training label is
the corresponding binary mask. In contrast, the training label is the edge aggregation point obtained by sampling the target boundary for the keypoint detection task.

Figure 1 shows the overall architecture of our model. In our pipeline, the box head and mask head are standard components of the original Mask R-CNN. We can simultaneously perform keypoint joint training with other heads by adding the keypoint detection branch. Suppose that we utilize \( k \) edge aggregation points and the centre point of the object, a total of \( k + 1 \) points, as the keypoint label information for training. And then, 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. As for 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, for capturing 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 to a single heat map. And then reduce the heat map to the same spatial scale as the mask prediction branch output by applying a \( 3 \times 3 \) convolution layer with a stride of 2. In the end, we broadcast the edge information of the instance to each channel of the mask prediction branch via feature fusion. Consequently, our model can pay more attention to the instance edge through the auxiliary prediction of the object boundary point.

The above fusion process could be mathematically described as follows:

\[
O_k = \sum_{i=1}^{k+1} \text{Conv}_{3 \times 3} \left( f_{k+1,i} \right),
\]

\[
O_m = O_k \odot f_m,
\]

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

### 3.5 Loss function

In original Mask R-CNN, the multi-task loss is defined as three parts: \( L_{\text{cls}} \), \( L_{\text{box}} \), and \( L_{\text{mask}} \). Specifically, for \( L_{\text{cls}} \), \( L_{\text{mask}} \), and \( L_{\text{keypoint}} \), we all use the cross-entropy loss \( L_{\text{CE}} \) function, while for \( L_{\text{box}} \) we use \( L_1 \) Loss. After the keypoint detection is added to our model, the loss function can be updated as follows:

\[
L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}} + \alpha L_{\text{keypoint}},
\]

\[
L_1(x, y) = \{L_1, \ldots, L_n\}^T (L_n = |x_n - y_n|),
\]

\[
L_{\text{CE}}(x, y) = -\frac{1}{n} \sum_x [y \ln(a) + (1 - y) \ln(1 - a)],
\]

where \( \alpha \) represents the weight parameter to balance the loss of the keypoint detection task. In Equations (4) and (5), \( x \) represents the input sample, \( y \) represents the label, \( a \) represents the predicted output, and \( n \) represents the total number of samples.
4 | EXPERIMENTS

In this section, we first introduce our experimental dataset, evaluation criteria, and hyper-parameters settings. Then we conduct ablation experiments from two aspects: the edge feature fusion and the influence of keypoint. Moreover, we report the results of instance segmentation only using the edge features of the keypoint.

4.1 | 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 of a fixed resolution of 2048 x 1024, and is split into 2975, 500, and 1525 images for train/val/test. In addition, the test dataset labels are not publicly released. We evaluate our results based on AP and AP50 metrics. Besides, we conducted plenty of ablation experiments on the validation set and provided results on the test dataset by uploading our results to the online server. Moreover, in order to verify that our method is not limited to a specific dataset, we perform experiment on COCO dataset to verify the generalization of our method. COCO dataset is regarded as one of the most watched and authoritative competitions in the field of computer vision. And it consists of 80 classes of common objects in daily life and provides 115k images for training and 5k images for verification, and 20k images for testing. Simultaneously, the test data set is also undisclosed, so we need to upload the test results to the server to evaluate the performance of our method.

4.2 | Implementation details

We perform our experiments based on the torchvision detection module with a pytorch backend. 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. We randomly horizontal flip the input image with a probability of 0.5 to augment the dataset. We train our model a total of 64 epochs, with a learning rate of 0.005 for the first 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 is taken as the initial model on this dataset, if not specially noted. And for keypoint detection tasks we use uniform sampling with 100 sampling points. For the COCO dataset we train our model for 24 epochs, with an initial learning rate of 0.005, and reduce it by 0.1 after 16 and 22 epochs, respectively.

4.3 | Main results on Cityscapes dataset

In this section, we compare our model with other state-of-the-art methods on the Cityscapes test subset, and the results are shown in Table 1. Training on “fine-only” data, our method outperforms Mask R-CNN by 3.8% on the validation subset and 2.7% on the test subset. More aggressively, we can obtain performance gains of 3.3% and 4.6% for the “people” and “car” categories, and we infer that this is because the two categories have relatively regular geometric shapes. Simultaneously, from the example segmentation results shown in the middle row of Figure 3, we can see that the keypoint auxiliary task we designed can effectively detect the contour of the object.

4.4 | Ablation study

4.4.1 | Edge feature fusion

The structural design choices for feature fusion: In our framework, feature fusion between multiple tasks is required, but the spatial scale of output tensor between instance segmentation task and keypoint detection task is different, that is, 56 x 56 versus 28 x 28. Therefore, we need to ensure that the input features of the fusion operation have the same spatial scale. There are a few design choices shown in Figure 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 channel-wise addition to the keypoint predictor output, the feature map spatial scale is reduced to $28 \times 28$ by using a $3 \times 3$ convolution layer with a stride of 2.

(b) The output of the keypoint predictor and mask predictor is the same as that of design (a). Before we do channel-wise addition on the keypoint predictor output, the feature map spatial scale is reduced to $28 \times 28$ by using a $3 \times 3$ convolution layer with a stride of 2.
TABLE 1 Results on Cityscapes val subset (denoted as AP [val]), and test subset (denoted as AP). * means our re-implementation, “fine-only” means only use Cityscapes dataset, “fine-only+coco” means pre-trained on coco dataset.

| Methods                        | AP [val] | AP | AP$_{50}$ | Person | Rider | Car | Truck | Bus | Train | Motorcycle | Bicycle |
|-------------------------------|----------|----|-----------|--------|-------|-----|-------|-----|-------|------------|---------|
| PaFPN (Li et al., 2022)       | -        | 28.3 | 47.3      | -      | -     | -   | -     | -   | -     | -          | -       |
| CenterPoly (Perreault et al., 2021) | -        | 15.5 | 39.4      | -      | -     | -   | -     | -   | -     | -          | -       |
| PolygonRNN++ (Acuna et al., 2018) | -        | 25.4 | 45.4      | 29.3   | 21.7  | 48.2| 21.1  | 32.3| 23.7  | 13.6       | 13.6    |
| Bshapenet+ [fine-only]        | -        | 27.3 | 50.4      | 29.7   | 23.3  | 46.7| 26.0  | 33.3| 24.8  | 20.3       | 14.1    |
| Neven et al. (2019)           | -        | 27.6 | 50.8      | 34.5   | 26.0  | 52.4| 21.6  | 31.1| 16.3  | 20.0       | 18.8    |
| GMIS (Liu, Yang, et al., 2018) | -        | 27.6 | 44.6      | 29.2   | 24.0  | 42.7| 25.3  | 37.2| 32.9  | 17.5       | 11.8    |
| BMask R-CNN (Cheng et al., 2020) | -        | 29.4 | 54.7      | 34.3   | 25.6  | 52.6| 24.2  | 35.1| 24.5  | 21.4       | 17.1    |
| GAI-N (Wu et al., 2020)       | -        | 32.0 | 59.4      | 36.0   | 29.0  | 52.8| 29.6  | 39.7| 28.9  | 23.2       | 18.5    |
| ICANet (Shang et al., 2022)   | -        | 32.1 | 58.2      | 37.5   | 28.5  | 56.4| 27.9  | 37.3| 26.4  | 22.9       | 20.5    |
| UPSNet (Xiong et al., 2019)   | -        | 33.0 | 59.6      | 35.9   | 27.4  | 51.8| 31.7  | 43.0| 31.3  | 23.7       | 19.0    |
| Axial-DeepLab-L (Wang et al., 2020) | -        | 33.2 | 54.9      | 31.9   | 27.8  | 52.3| 30.6  | 44.1| 33.9  | 25.7       | 19.2    |
| LevelSet R-CNN (Homayounfar et al., 2020) | -        | 33.3 | 58.2      | 36.9   | 29.2  | 54.6| 30.4  | 39.3| 30.2  | 25.4       | 20.3    |
| S3B-Net (Zhang, Li, et al., 2021) | -        | 33.6 | 61.2      | 37.8   | 30.3  | 55.2| 30.6  | 40.9| 28.7  | 21.3       | 20.7    |
| Axial-DeepLab-XL (Wang et al., 2020) | -        | 33.9 | 55.8      | 32.3   | 28.1  | 52.6| 32.6  | 41.7| 38.0  | 25.8       | 20.4    |
| E2EC (Zhang et al., 2022)     | 34.0     | 30.3 | 54.0      | 40.7   | 27.9  | 55.4| 28.4  | 35.8| 20.1  | 20.9       | 13.2    |
| Temporal (Zhu et al., 2022)   | 30.9     | -    | -         | -      | -     | -   | -     | -   | -     | -          | -       |
| Mask R-CNN [fine-only] (He et al., 2020) | 31.5     | 26.2 | 49.9      | 30.5   | 22.7  | 46.9| 22.8  | 32.2| 18.6  | 19.1       | 16.0    |
| Mask R-CNN* [fine-only] (He et al., 2020) | 33.1     | 28.5 | 55.0      | 34.0   | 25.7  | 51.3| 24.4  | 32.1| 22.1  | 20.9       | 17.5    |
| 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    |
| Ours [fine-only+coco]         | -        | 34.3 | 59.3      | 40.0   | 30.8  | 57.5| 30.4  | 40.8| 28.6  | 25.2       | 20.9    |

FIGURE 3 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.

(c) The output of the keypoint predictor is the size of $56 \times 56 \times (k+1)$, and the output of the mask predictor is the size of $28 \times 28 \times C$. We upsample the spatial output scale of the mask predictor to $56 \times 56$ using transposed convolutional layer.
(d) The output of keypoint predictor is size of $28 \times 28 \times (k+1)$, and the output of mask predictor is size of $28 \times 28 \times C$. 

(c) The output of the keypoint predictor is the size of $56 \times 56 \times (k+1)$, and the output of the mask predictor is the size of $28 \times 28 \times C$. We upsample the spatial output scale of the mask predictor to $56 \times 56$ using transposed convolutional layer.
The experimental results are shown in Table 2. We can see that each method can achieve performance improvement except for method (a). In method (a), we only use one convolution layer to process the prediction results with a single channel, which will lose a lot of edge prediction information. Comprehensive performance comparison, we chose option b as our design choice.

The spatial scale reduction designs 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: max pooling, average pooling, and a $3 \times 3$ convolution layer with a stride of 2.

Table 3 shows the results for the above spatial scale reduction designs, and we can see that all three methods can improve the performance of the model, which further validates the effectiveness of our approach. Notably, we can see that compared with a simple pooling operation, for
our model, using a convolutional downsampling method with learnable parameters performs better, so we finally adopted the convolution method for downsampling.

**The choices of the fusion mode:** After channel-wise addition and spatial scale reduction of the keypoint head output, the following operation is feature fusion. In our experiment, we test three methods of maximum, multiplication, and addition to choosing the best fusion strategy. Results are shown in Table 4. We can see that the fusion method of multiplication can be better adapted to our model, so we choose this mode as our fusion strategy.

### 4.4.2 Influence of keypoint

The model we designed simultaneously requires multiple training tasks, including box head, mask head, and keypoint head. But the Cityscapes dataset only provides instance ground truths masks, so we need to build the bounding box and edge aggregation point labels from the ground truths masks ourselves. For a single target, the bounding box label is uniquely fixed, but we need to sample it from the object's edge for the contour point label, which means that we have more alternative strategies. Specifically, we separately analyze the impact of corner sampling and uniform sampling on our method, and the results are shown in Table 5. Analogously, corner sampling reflects the essential geometric information of the object, whereas uniform sampling reflects the general geometric details of the object. From the experimental results, we can see that it is more conducive to instance segmentation by using keypoint detection technology to capture the overall geometric position information of the object. Therefore, we used the average sampling method to make our labels.

Furthermore, we demonstrate the influence of different sampling points number on the model's performance. The results are shown in Table 6. Specifically, we conduct experiments on 50 and 100 sampling points, respectively. And we find that the further increase of the sampling points number will damage the model's effectiveness. The reason is that too many sampling points will increase the parameter of the keypoint head, which will cause the network to be more prone to overfitting.

Moreover, following the intuition that the model can learn the geometric relationship between the object boundary point and the centre point, we use not only the points sampled from the object contour but also the centre point of the object. Since the centre point of the object can be regarded as the origin in a polar coordinate system, and thus the distance and angle of the object boundary point relative to the centre point

### TABLE 3 Results of different spatial scale reduction designs 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  |
| $3 \times 3$ convolution     | 36.0| 63.0  |

Note: Values highlighted in bold represent the optimal value for each column.

### TABLE 4 Results of different fusion strategies 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  |

Note: Values highlighted in bold represent the optimal value for each column.

### TABLE 5 Results of different contour sampling methods 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  |

Note: Values highlighted in bold represent the optimal value for each column.
have certain regularity for the objects in the same category. We evaluate the effect of adding an object centre point using COCO evaluation metrics to verify our analysis. As shown in Table 7, where $AP_{bb}$ represents the AP value of the bounding box detection results, $AP_{mask}$ and $AP_{keypoint}$ represent the AP value of the mask predict outcomes and keypoint estimation results, respectively. And the subscripts $s$, $m$, and $l$ represent small, medium, and large objects, respectively. We can see that our method can simultaneously improve the performance of instance segmentation, keypoint estimation, and object detection tasks.

### 4.4.3 Influence of the balance coefficient $\alpha$

In our model, keypoint detection is only an auxiliary task, and we hope that it can complete its duties while not affecting other tasks as much as possible. Therefore, we balance the importance of each task in the multi-task joint training by setting the attenuation coefficient of its loss function. The experimental results are shown in Table 8. From the results in the table, we can see that our method is susceptible to the set balance coefficient, and we can get the best performance when we set $\alpha=0.5$.

### 4.5 Experimental results on COCO dataset

Compared with the Cityscapes dataset, the COCO dataset contains objects of more classes and scales, so it is more challenging for our algorithm. We experimentally set the loss balance weight coefficient $\alpha=0.1$ on the COCO dataset. The experimental results are shown in Table 9. Notably, our method achieves noticeable performance gains in instance segmentation and object detection tasks compared with baseline with a slight increase in forwarding inference time. Moreover, in Table 10 we report the comparison results of our method with state-of-the-art methods on the COCO dataset. Compared with the Mask R-CNN baseline, our method achieves significant performance improvements on other evaluation metrics except for a slight degradation on small objects. And this performance difference on small
objects we infer is caused by the keypoint sampling strategy we adopt. As described in Section 3.3, when the number of small object contour points sampled is insufficient, we will try to obtain a fixed number of sampling points by randomly extracting existing contour points to meet the input requirements of the deep neural network. And this way of randomly selecting repeated sampling points may harm the judgment of the network.

4.6 | Quantitative analysis

Moreover, to verify the effectiveness of our method more intuitively, we visualize the final feature map obtained by the keypoint head and the corresponding feature map obtained by the mask head. Specifically, we normalize the two feature maps and generate related heat maps. Some examples are demonstrated in Figure 5 (the darker pixels in the heat map represent higher scores). From these examples, we can infer that the output feature maps of keypoint head can learn the boundary information of the object and get comprehensive background information. Therefore, the final output feature of the mask head obtained by fusing the two tasks is equivalent to segmenting the object by combining the background and foreground features. The inward activation and outward activation in Figure 5 are driven by the different supervisory functions of the subtasks in the multi-task joint training. In Figure 5, the heatmaps shown in columns 2 and 4 are obtained from the mask head. The supervised label of the mask head loss function is a binary mask used to classify the foreground and background of the object. Therefore, its prediction aims to activate the area where the object is located and suppress the background information through the mask head neural sub-network, resulting in inward activation. In contrast, the heatmaps shown in columns 1 and 3 are generated from the newly added object contour keypoint detection sub-branch, which is used to facilitate the model’s attention to object boundaries. From the activation map, we can see that the contour keypoint detection sub-network can highlight the object's boundary information and further drive the activation of the overall background of the object. We infer that this is raised by the gradient interaction flow of multi-task joint training.

Besides, we show some examples of our method and Mask R-CNN baseline in Figure 6. It can be seen from the figures that, compared with the baseline, our approach can fit the contour of the object better, which is thanks to our newly added keypoint detection auxiliary task and effective fusion strategy.

| Methods          | Backbone         | AP<sup>bb</sup> | AP<sup>bb</sup><sub>s</sub> | AP<sup>bb</sup><sub>m</sub> | AP<sup>bb</sup><sub>f</sub> | AP<sup>mask</sup> | AP<sup>mask</sup><sub>s</sub> | AP<sup>mask</sup><sub>m</sub> | AP<sup>mask</sup><sub>f</sub> | Time (ms/img.) |
|------------------|------------------|-----------------|------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Mask R-CNN       | ResNet-50-FPN    | 37.9            | 21.7             | 41.3           | 49.6           | 34.6           | 15.8           | 37.2           | 51.1           | 59.3           |
| Ours             | ResNet-50-FPN    | 38.3            | 21.3             | 41.8           | 50.3           | 35.4           | 15.7           | 38.4           | 51.9           | 65.2           |

Note: Values highlighted in bold represent the optimal value for each column.

| Methods                      | Backbone         | AP<sup>mask</sup> | AP<sup>mask</sup><sub>s</sub> | AP<sup>mask</sup><sub>m</sub> | AP<sup>mask</sup><sub>f</sub> |
|------------------------------|------------------|-------------------|------------------|----------------|----------------|
| MS YOLACT (Zeng et al., 2022) | ResNet-101-FPN  | 30.0              | 10.2             | 31.5           | 48.3           |
| RISAT (Pei et al., 2022)     | ResNet-101-FPN  | 31.9              | 8.8              | 32.7           | 53.0           |
| YOLACT (Bolya et al., 2019)  | ResNet-101-FPN  | 31.2              | 12.1             | 33.3           | 47.1           |
| PolarMask (Xie et al., 2020) | ResNet-101-FPN  | 32.1              | 14.7             | 33.8           | 45.3           |
| PolarMask ++ (Xie et al., 2021) | ResNet-101-FPN | 33.8              | 16.6             | 35.8           | 46.2           |
| BlendMask (Chen et al., 2020)| ResNet-50-FPN   | 34.3              | 14.9             | 36.4           | 48.9           |
| CDANet (Wang et al., 2022)   | ResNet-50-FPN   | 34.4              | 17.8             | 36.5           | 45.5           |
| DANCE (Liu, Liew, et al., 2021)| ResNet-50-FPN | 34.6              | 19.3             | 37.2           | 43.9           |
| SipMaskv2 (Cao et al., 2022) | ResNet101-FPN  | 34.7              | 13.2             | 37.8           | 52.8           |
| SparseInst (Cheng et al., 2022)| ResNet-50-FPN | 34.7              | 14.3             | 36.2           | 50.7           |
| Mask R-CNN (He et al., 2020) | ResNet-50-FPN   | 34.9              | 19.0             | 37.4           | 45.0           |
| BorderPointsMask (Yang et al., 2022) | ResNet-101-FPN | 35.0              | 17.1             | 37.4           | 48.6           |
| CondInst (Tian, Shen, & Chen, 2020) | ResNet-50-FPN | 35.4              | 18.4             | 37.9           | 46.9           |
| Mask Scoring R-CNN (Huang et al., 2019) | ResNet-50-FPN | 35.8              | 16.2             | 37.4           | 51.0           |
| Ours                        | ResNet-50-FPN   | 35.7              | 18.6             | 38.1           | 46.4           |
In this section, we analyze the effectiveness of our method from the view of multi-task joint training and introduce our future work.

Mask R-CNN has carried out multi-task joint training of object detection, keypoint estimation, and instance 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. In contrast, both the performance of the detection and segmentation tasks is degraded (as shown in Table 7). We deduce that this is because in the task of human pose estimation, the labels of keypoints are located within the human body, and the overall information of the instances captured by the mask head network can help locate these keypoints. But for the object detection task, including positioning and classification, and the instance segmentation task composed of pixel-wise classification, the keypoint information does not help. In our method, the task of keypoint auxiliary detection is to obtain the contour boundary of the object, so its purpose is the same as the segmentation task of Mask Head. And after adding the auxiliary detection of the object centre point and adjusting the loss balance coefficient, our method can improve the performance of object bounding box detection at the same time. Besides, in the joint training, we can get the edge point information of the target through the keypoint estimation task. In another way, we can segment the object by only using the contour information mapped by these edge points. We carried out experiments to verify its performance, and when $k = 100$, we can get an AP value of 12.5 on the Cityscapes val subset.

In addition, our work still has some problems that need to be solved. First, compared to the original Mask R-CNN, the newly added keypoint head has more parameters, increasing the computational burden. Therefore, we need to explore more lightweight designs to optimize our methods in the future. And as mentioned above, our model has a strong sensitivity to the setting of the balance coefficient $\alpha$ in the multi-task joint training with different datasets, which is also a direction that needs further research.

Although we can use lightweight convolution methods such as MobileNet (Howard et al., 2017), ShuffleNet (Zhang et al., 2018), and GhostNet (Han et al., 2020) to slightly reduce the model’s dependence on computing resources, only simple convolutional layer replacement in the keypoint detection sub-network will lead to the degradation of model performance. In future work, we intend to simultaneously optimize the

---

**FIGURE 5** Examples of the instance heat map. The first and third columns are the output feature maps of the keypoint head; the second and fourth columns are the output feature maps of the mask head.
detection of keypoint auxiliary tasks and the original mask head process to achieve a better accuracy-complexity balance in the overall architecture.

As for the sensitivity of the $\alpha$ coefficient to different datasets, we plan to use predictable adaptive coefficients to replace the current dataset-specific coefficients in the future. The schemes mentioned above require more elaborate design and research, so we will conduct further research in future work and not discuss them in detail in this article. Our proposed object contour keypoint detection auxiliary task can successfully enhance the instance segmentation model's attention to object edges, enabling more delicate segmentation boundaries. Simultaneously, we introduce a coefficient $\alpha$ into the joint loss function to balance the importance of multi-task joints, which can highlight the role of auxiliary tasks while reducing the impact on other existing tasks.

6 | CONCLUSION

In this article, 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 experiments on the Cityscapes dataset and COCO dataset have proved the effectiveness of our method. We hope that our method can have some inspiration for other instance segmentation methods.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 62032013) and the Fundamental Research Funds for the Central Universities (No. N2324004-12).
CONFLICT OF INTEREST STATEMENT
The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT
Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID
Chong Fu https://orcid.org/0000-0002-4549-744X
Chiu-Wing Sham https://orcid.org/0000-0001-7007-6746

REFERENCES
Acuna, D., Ling, H., Kar, A., & Sidler, S. (2018). Efficient interactive annotation of segmentation datasets with polygon-RNN++. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 859–868). IEEE.

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS—improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision (pp. 5561–5569). IEEE.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 9157–9166). IEEE.

Cao, J., Pang, Y., Anwer, R. M., Cholakkal, H., Khan, F. S., & Shao, L. (2022). SipMaskv2: Enhanced fast image and video instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45, 3798–3812.

Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., & Yan, Y. (2020). BlendMask: Top-down meets bottom-up for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8573–8581). IEEE.

Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C. C., & Lin, D. (2019). Hybrid task cascade for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4974–4983). IEEE.

Chen, L., Strauch, M., & Merhof, D. (2019). Instance segmentation of biomedical images with an object-aware embedding learned with local constraints. In D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, & A. Khan (Eds.), International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 451–459). Springer.

Chen, L.-C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., & Adam, H. (2018). MaskLab: Instance segmentation by refining object detection with semantic and direction features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4013–4022). IEEE.

Cheng, T.-G., Wang, X., Chen, S., Zhang, W., Zhang, Q., Huang, C., & Liu, W. (2022). Sparse instance activation for real-time instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4433–4442). IEEE.

Cheng, T.-G., Wang, X., Huang, L., & Liu, W. (2020). Boundary-preserving mask R-CNN. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), European Conference on Computer Vision (pp. 660–676). Springer.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3213–3223). IEEE.

Ding, H., Jiang, X., Liu, A. Q., Thalmann, N. M., & Wang, G. (2019). Boundary-aware feature propagation for scene segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6819–6829). IEEE.

Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P., & Qian, C. (2020). CentripetalNet: Pursuing high-quality keypoint pairs for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10519–10528). IEEE.

Gao, N., Shan, Y., Zhao, X., & Huang, K. (2021). Learning category-and instance-aware pixel embedding for fast panoptic segmentation. IEEE Transactions on Image Processing, 30, 6013–6023.

Geng, Z., Sun, K., Xiao, B., Zhang, Z., & Wang, J. (2021). Bottom-up human pose estimation via disentangled keypoint regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14676–14686). IEEE.

Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440–1448). IEEE.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). GhostNet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1580–1589). IEEE.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42, 386–397.

Homayounfar, N., Xiong, Y., Liang, J., Ma, W.-C., & Urtasun, R. (2020). LevelSet R-CNN: A deep variational method for instance segmentation. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), European Conference on Computer Vision (pp. 555–571). Springer.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Huang, Z., Huang, L., Gong, Y., Huang, C., & Wang, X. (2019). Mask scoring R-CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6409–6418). IEEE.

Jiang, J., Ji, Y., Wang, X., Liu, Y., Wang, J., & Long, M. (2021). Regressive domain adaptation for unsupervised keypoint detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6780–6789). IEEE.

Kang, B. R., Lee, H., Park, K., Ryu, H., & Kim, H. Y. (2020). BshapeNet: Object detection and instance segmentation with bounding shape masks. Pattern Recognition Letters, 131, 449–455.

Ke, W., Zhang, T., Huang, Z., Ye, Q., Liu, J., & Huang, D. (2020). Multiple anchor learning for visual object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10206–10215). IEEE.

Khalfaï, N. E. M., Manogaran, G., Taha, M. H. N., & Loey, M. (2021). A deep learning semantic segmentation architecture for COVID-19 lesions discovery in limited chest CT datasets. Expert Systems, 39, e12742.

Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., & Rother, C. (2017). InstanceCut: From edges to instances with multitask. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5008–5017). IEEE.
AUTHOR BIOGRAPHIES

**Wenchao Zhang** received the M.S. degree in Communication and Information System from the Shenyang University of Science and Engineering, Shenyang, China. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Northeastern University, Shenyang, China. His research interests include computer vision, machine learning, and in particular, on instance segmentation.

**Chong Fu** received the Ph.D. degree in Computer Software and Theory from Northeastern University, Shenyang, China, in 2001 and 2006, respectively. He joined Northeastern University in 2001, where he is currently a Professor with the Department of Communication and Electronics Engineering, School of Computer Science and Engineering. In 2010, he spent 3 months as a Visiting Researcher with the Department of Electronics Information Engineering, The Hong Kong Polytechnic University. His research interests include multimedia security and computer vision.

**Mai Zhu** received the B.E. degree in Electronic and Information Technology as well as the M.S. degree in Computer Technology from Northeastern University, Shenyang, China. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Northeastern University, Shenyang, China. Her research interests include computer vision and machine learning, especially deep learning.

**Lin Cao** received the B.Eng. degree in telecommunication engineering from Northeastern University, Shenyang, China, in 1999, and the Ph.D. degree in signal and information processing from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with the Department of Electronic Engineering, Beijing Information Science and Technology University (BISTU), Beijing. He teaches courses on digital signal processing, digital image processing, and soft design fundamentals. He is also the Dean of the School of Information and Communication Engineering and the Deputy Director of the Key Laboratory of the Ministry of Education for Optoelectronic Measurement Technology and Instrument, BISTU. He has published over 40 articles on image processing and pattern recognition. His research interests include radar signal processing and image understanding and recognition. Dr. Cao is a member of the Institute of Electronics, Information and Communication Engineers (IEICE), and the China Education Society of Electronics.

**Ming Tie** received the Ph.D. degree from Northeastern University, Shenyang, China in 2006. From 2007 to 2008, he was a Post-doctoral fellow in Tsinghua University Beijing. He joined Beijing Institute of Nearspace Vehicle's System Engineering, Beijing, in 2009, where he is currently a professor. His current research interests include intelligent modelling and simulation of complex systems, designing and simulation of intelligent nearspace vehicles.

**Chiu-Wing Sham** received the Bachelor degree (Computer Engineering) and MPhil. degree from The Chinese University of Hong Kong in 2000 and 2002 respectively, and received the Ph.D. degree from the same university in 2006. He began his research work on digital design during the final year project as an undergraduate which focused on improving the performance, reducing the logic complexity of the system and, hence, power consumption. He has worked as an Electronic Engineer on the FPGA applications of motion control system and system security with cryptography in ASM Pacific Technology Ltd (HK). During the years at the Hong Kong Polytechnic University, he has also
engaged in various university projects for the commercialization of technology, in particular, a few optical communication projects which are in collaboration with Huawei. He also worked on physical design of VLSI design automation. He was invited to work at Synopsys, Inc (Shanghai) in the summer of 2005 as a Visiting Research Engineer. He also obtained the Best Paper Award in ISQED 2013 and the Best Paper Award in ATC 2015. He is an IEEE Senior Member and now working at The University of Auckland as Senior Lecturer.

How to cite this article: Zhang, W., Fu, C., Zhu, M., Cao, L., Tie, M., & Sham, C.-W. (2024). Joint object contour points and semantics for instance segmentation. *Expert Systems*, 41(3), e13504. [https://doi.org/10.1111/exsy.13504](https://doi.org/10.1111/exsy.13504)