

Article

Image Semantic Segmentation Use Multiple-Threshold Probabilistic R-CNN with Feature Fusion

Jianxin Liu , Yushui Geng *, Jing Zhao, Kang Zhang  and Wenxiao Li

School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China; 1043118431@stu.qlu.edu.cn (J.L.); zj@qlu.edu.cn (J.Z.); 1043118447@stu.qlu.edu.cn (K.Z.); 1043118412@stu.qlu.edu.cn (W.L.)

* Correspondence: gys@qlu.edu.cn

Abstract: With continuous developments in deep learning, image semantic segmentation technology has also undergone great advancements and been widely used in many fields with higher segmentation accuracy. This paper proposes an image semantic segmentation algorithm based on a deep neural network. Based on the Mask Scoring R-CNN, this algorithm uses a symmetrical feature pyramid network and adds a multiple-threshold architecture to improve the sample screening precision. We employ a probability model to optimize the mask branch of the model further to improve the algorithm accuracy for the segmentation of image edges. In addition, we adjust the loss function so that the experimental effect can be optimized. The experiments reveal that the algorithm improves the results.

Keywords: deep learning; image semantic segmentation; multiple threshold



Citation: Liu, J.; Geng, Y.; Zhao, J.; Zhang, K.; Li, W. Image Semantic Segmentation Use Multiple-Threshold Probabilistic R-CNN with Feature Fusion. *Symmetry* **2021**, *13*, 207. <https://doi.org/10.3390/sym13020207>

Received: 25 December 2020

Accepted: 25 January 2021

Published: 28 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Image segmentation is a crucial image preprocessing method in computer vision, and it is a classic problem in computer vision research. Image segmentation refers to dividing an image into several nonoverlapping subregions so that the features in the same subregion have a certain similarity, and the features in different subregions exhibit obvious differences. Image segmentation has many applications, such as augmented reality, autonomous driving, human–computer interaction, and video content analysis. The goal is to assign a category label to each pixel to provide comprehensive scene understanding. In practical problems, many application scenarios must process a large volume of image data simultaneously, and the image types are complex. Traditional image segmentation algorithms, such as threshold-based segmentation algorithms and watershed algorithms can no longer meet the current needs. With the rapid development of deep learning, deep learning solutions are increasingly applied to computer vision.

Many image segmentation algorithms are based on deep learning, such as VGGNet [1] and ResNet [2], which are very good image segmentation algorithms. Thus far, these two networks still have extremely high dominance in the field of feature extraction. Long et al. [3] proposed the Full Convolution Network (FCN) published on CVPR in 2015, which has become an industry benchmark in the field of image segmentation. The CNN convolutional layer is connected to the fully connected layer. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Many image segmentation methods use the FCN or a part of it. Pinheiro et al. [4] proposed a deep mask segmentation model, which segments each instance object by outputting a prediction candidate mask through instances displaying in the input image, but the accuracy of the boundary segmentation was low. He et al. [5] proposed a Mask R-CNN framework, which is a better algorithm for instance segmentation among the existing segmentation algorithms. Huang et al. [6] addressed the problem that the Mask R-CNN classification box and prediction mask share the evaluation function, which interferes with the segmentation

results, and proposed the Mask Scoring R-CNN to optimize the information transmission of the former and improve the quality of the generated prediction mask. After training with large quantities of data, he won the COCO 2017 challenge instance segmentation task championship.

Compared with other image segmentation algorithms, the image segmentation model Mask Scoring R-CNN not only exhibits considerable improvement in segmentation accuracy but also has strong robustness and is superior in the segmentation of small target images. It is widely used in agriculture, construction, transportation, medical image segmentation, and other fields. Tian et al. [7] used the improved Mask Scoring R-CNN to segment the image of apple blossoms to judge the growth of an apple tree. The recognition accuracy rate reached 96.43%, and the recall rate reached 95.37%. Zhang et al. [8] proposed a method based on Mask R-CNN, and the improved algorithm combined with Mask Scoring R-CNN is used for traffic monitoring to obtain comprehensive vehicle information, such as vehicle type, number of axles, speed, length, current driving lane, and traffic volume. In three test videos, for the average number of models and axles, the recognition accuracy rates were above 97% and 88%, respectively. A speed error of more than 90% of the vehicles was less than 4%. Liu et al. [9] improved Mask Scoring R-CNN to achieve a high performance in medical image segmentation, which is a good improvement compared with the current state-of-the-art methods.

Thus, we propose a new image segmentation algorithm based on multiple thresholds to obtain a better mask prediction result for the Mask Scoring R-CNN algorithm. The main contributions of this work are as follows:

- We optimize the feature extraction of the network. Based on the feature pyramid, we propose a feature fusion method, combining two feature maps of different sizes to obtain a new feature map.
- We propose a multiple threshold segmentation architecture as the segmentation branch of the network. A multiple threshold architecture can better filter redundant information. The integration of a Dense Conditional Random Field (DenseCRF) improves the network performance further.
- We evaluate this method on the Microsoft Common Objects in Context (MS COCO) [10] dataset, including ablation experiments. In Figure 1, we show a demo of network output. The results reveal that our network achieved good results on semantic segmentation, with an AP reaching 40.9.



Figure 1. A demo of our network output. The left is the original image, and the right is the network output. Masks are shown in color curve, bounding boxes are shown in rectangle, category and confidences are also shown.

In the rest of this article, we firstly introduce the related work in Section 2. In Section 3, we will introduce the structure of the network in detail. In addition, Section 4 presents the experiment and results.

2. Related Work

The work in this paper is based on previous work, including the Mask Scoring R-CNN. This section introduces the previous work.

2.1. Mask Scoring R-CNN

The Mask Scoring R-CNN [6] is an excellent image segmentation algorithm published in 2019. It is an improved method of the Mask R-CNN. It surpasses the mask R-CNN in the task of COCO image instance segmentation without large-scale data training. The key to its surpassing its predecessors is the scoring. The prediction mask and classification confidence in the Mask R-CNN share the scoring function, both of which calculate the score for the classification confidence of the target area. However, there is no strong correlation between the two, resulting in lower quality and accuracy of the predicted mask. Therefore, the Mask Scoring R-CNN uses a dedicated network module to learn and predict the quality of the mask. This module uses instance features and the prediction mask as input. It evaluates the intersection of union (IoU) value between the prediction mask and ground truth, and then the IoU is used as a metric for the prediction mask. Hence, the Mask Scoring R-CNN can obtain more accurate segmentation results than the Mask R-CNN.

2.2. Cascade R-CNN

This model [11] solves the problem that the detection frame is not particularly accurate in target detection and is prone to noise interference. The experiment found that, in the anchor-based detection method, because we generally set the positive and negative samples for training (for training classification and coordinate regression of the positive samples), and the method of selecting the positive and negative samples primarily uses the IoU between the candidate box and ground truth. The commonly used IoU ratio is 50% (i.e., an IoU of greater than 0.5 is used as a positive sample, and an IoU of less than 0.3 is used as a negative sample). However, the question is whether the threshold value of 0.5 is best.

The experiments indicate that setting a high threshold will yield better results. However, if the threshold is too high, overfitting occurs. No matter how the threshold is set, the network has a certain optimization effect on the input proposal. Based on these two points, the author designed the Cascade R-CNN to iteratively process the network. Through the cascaded R-CNN, each cascaded R-CNN sets a different IoU threshold. So in each network, the accuracy of the output is improved and used as the input of the next higher-precision network, and the accuracy of the network output is gradually improved.

3. Methods

In this section, we will introduce the multiple-threshold probabilistic R-CNN (MTP R-CNN) network structure. The network mainly includes backbone, region proposal network (RPN), region of interest (RoI) Align, and three RoI head networks used to complete specific tasks. We show the overall structure of the network in Figure 2.

3.1. Backbone

The backbone of our network uses ResNet-101, where the number of network layers is 101. We compared ResNet-101 with ResNet-18 and ResNet-50 in the experiment. Because the images are diverse and complex, a single CNN alone cannot extract the image features well. Therefore, integrating a Feature Pyramid Network (FPN) [12] in the network can yield a better feature map. The FPN is a top-down hierarchical structure with horizontal connections, solves the multi-scale problem of extracting object feature from images. The structure has strong robustness and adaptability and requires fewer parameters. This paper uses an FPN structure to generate different-sized feature maps.

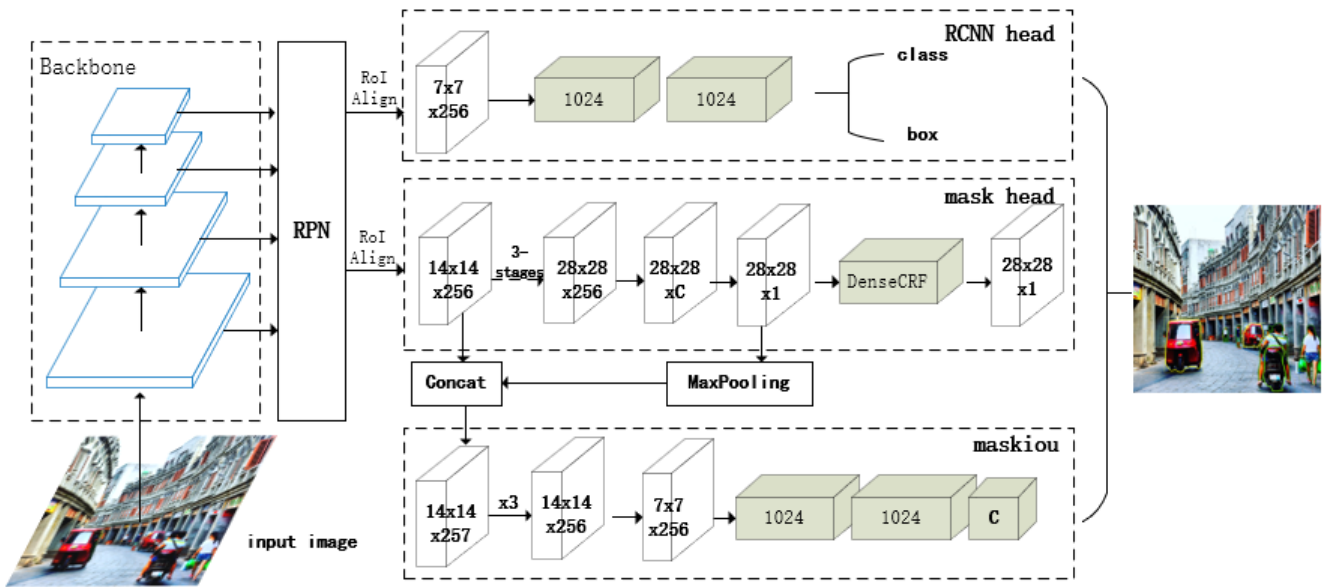


Figure 2. Multiple-threshold probabilistic R-CNN (MTP R-CNN) network structure. The backbone structure in this figure is simplified. The image enters the backbone as input, and RoIs are generated through region proposal network (RPN). RoIs enter the R-CNN head and mask head through Align layer to generate the final result.

Our backbone structure is presented in Figure 3, where C_2 to C_5 and P_2 to P_5 indicate a symmetrical structure. The feature map contains rich long-range relationship and location information. Our main contribution is reflected in the fusion module, which can be expressed as follows:

$$P_i = f(u(P_{i+1}) + g(C_i)). \quad (1)$$

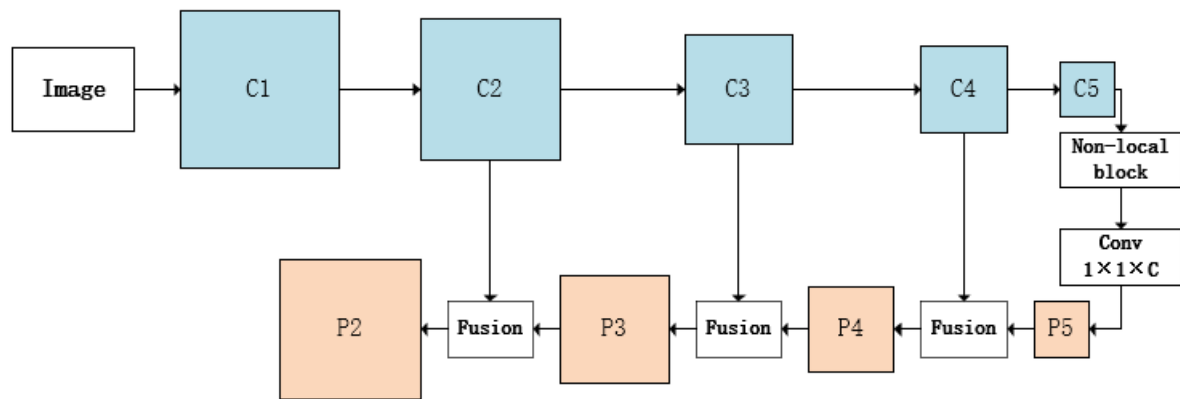


Figure 3. The structure of the backbone network. Finally, four feature maps of different sizes will be obtained.

Among them, the value range of i is $[2, 5]$, P_i represents the feature map of the feature pyramid network. Additionally, C_i represents feature maps of different scales generated by ResNet. The function $f()$ represents a 1×1 convolutional layer, the function $u()$ represents upsampling, and the function $g()$ represents non-local block. In particular, when $i = 5$, $P_i = f(g(C_i))$.

The structure of fusion module is displayed in Figure 4. First, the feature map of the higher P_i is upsampled with the spatial resolution by a factor of 2. The corresponding C_i feature map is connected to the upsampled map after the non-local block pass. Finally, through a 1×1 convolutional layer, the channel changes to the number of output channels we set. In particular, P_5 has no information transmitted by the upper layer, so we import C_5 into the non-local block and then change the number of channels through a 1×1

convolution layer to obtain P_5 . In this way, we attain four feature maps of different sizes, which are used as the input of the RPN network.

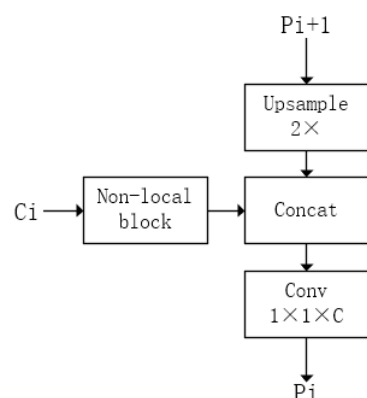


Figure 4. The structure of the fusion module. Upsample is the upsampling operation, concat represents the connection operation, and conv is a 1×1 convolutional layer.

Then, we introduce the non-local block [13]. The convolution unit in the CNN only focuses on the area of neighboring kernel size, ignoring the contribution of other global areas (such as distant pixels) to the current area. Thus, non-local blocks must capture this long-range relationship. For two-dimensional (2D) images, the relationship is the weight of any pixel in the image compared to the current pixel. Non-local neural networks are flexible building blocks, as illustrated in Figure 5. The network does not change the shape and size of the input to be easily used with convolutional layers. Moreover, the network can be added to the early part of the neural network, unlike the fully connected layer, which is usually used at the end. Non-local block allows us to build a richer hierarchy, combining non-local and local information.

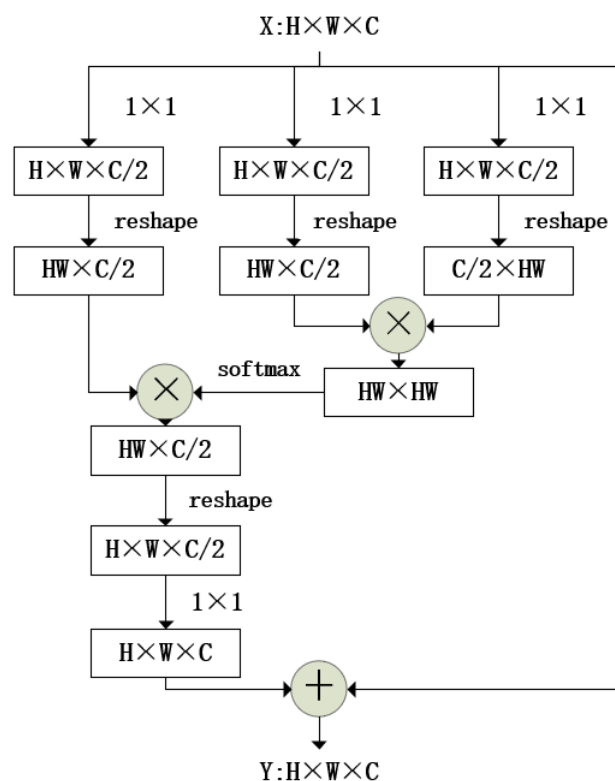


Figure 5. The structure of non-local block. 1×1 represents a 1×1 convolutional layer.

3.2. Region Proposal Network and Region of Interest Align Layer

Feature maps of different sizes generated by the previous network are input into the RPN [14]. The RPN is equivalent to a classless object detector based on a sliding window. It is based on the structure of a CNN. The sliding window scan produces anchors. A suggested pixel can generate numerous anchors of different sizes and aspect ratios. Then, the RPN extracts the features from different levels of the feature pyramid in the light of the size of the object. Thus, the detection performance of small targets is greatly improved without greatly increasing the amount of calculation. The accuracy and speed are also greatly improved.

Because our image segmentation is a pixel-level operation, it is necessary to determine whether a given pixel is part of the target for segmentation, and the accuracy must be at the pixel-level. After performing a series of convolution and pooling operations on the original image and a frame fine-tuning step in the RPN, the RoI box can have different sizes. If the pixel-level segmentation is directly performed, the image target object cannot be accurately positioned, so it must restore the RoI. Here, we use the RoI Align layer [5] in the Mask R-CNN network.

First, we select the feature map P_i that should be aligned according to the formula, and then use RoI Align layer to align to the selected feature map. The formula is as follows:

$$i = i_0 + \log_2 \left(\sqrt{wh} / 224 \right). \quad (2)$$

Here, 224 is the standard pre-training size, i_0 is the target level to which an RoI with $w \times h = 224^2$ should be mapped, and we set it to 4. The RoI Align layer uses bilinear interpolation to retain the spatial information on the feature map and perfectly solves the error caused by the two quantizations of the feature map in the RoI Pooling layer [15] and the problem of regional mismatch of image objects. In this way, pixel-level detection and segmentation can be realized.

3.3. Class Head, Bbox Head, and Mask Head

In the Mask Scoring R-CNN, the segmentation and detection branches are inserted in parallel. Since we need to set multiple thresholds. Considering that image segmentation is a pixel-level operation, the detection of instance objects is not very helpful for segmentation. Too many overlay networks can lead to slow operation, so only multi-level thresholds are set for the segmentation branch. For the classification branch and bbox regression branch, we use the configuration in the Mask Scoring R-CNN.

The threshold in this article means that in training, the head network filters RoIs based on the IoU threshold. We considered those RoIs as the positive sample whose IoU was larger than the threshold. These positive samples were used in subsequent network learning. We could only treat all RoIs as positive samples and let the head network inference because we did not know the groundtruth during inference.

Our main body uses the FCN for the mask branch, as is shown in Figure 6. The experimental results show that the higher the threshold, the better the predicted results. However, a higher threshold can sharply reduce the number of positive samples, leading to overfitting. If the threshold is too low, the samples would contain more redundant information. Therefore, it would be difficult for the segmentation network to distinguish between positive and negative samples, affecting the training effect. Hence, we propose a multi-threshold segmentation branch. A segmentation branch is added at each cascade stage, which maximizes the sample diversity for learning mask prediction tasks. According to the experiment and the experience of Cascade R-CNN, we found that setting the three-level threshold can achieve the best effect without losing too much performance. In this way, the result of the previous mask branch can pass through the Maxpolling layer and then connect with RoIs as the input of the next branch.

On this basis, the DenseCRF [16] is used to optimize the segmentation result. As mentioned, in the application of image segmentation, the CRF processes the edges of

image segmentation more delicately. Therefore, based on the original mask branch, the DenseCRF optimizes the segmentation results of the previous network to improve the final segmentation accuracy. Due to the characteristics of DenseCRF, we only access DenseCRF on the last-level mask branch.

To overcome the limitations of the short-range CRF, we modified the DenseCRF model. The model uses the following energy function:

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j), \quad (3)$$

where x is the label assigned to the pixel. We use it as a single point potential $\theta_i(x_i) = -\log P(x_i)$, where $P(x_i)$ is the probability of the label distribution at pixel i . A potential energy pair has the same form and can be inferred using a fully connected graph, for example, connecting all pairs of image pixels, i and j . We use the following expression:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[\omega_1 \exp\left(-\frac{\|P_i - P_j\|^2}{2\sigma_\alpha^2}\right) - \frac{\|C_i - C_j\|^2}{2\sigma_\beta^2} + \omega_2 \exp\left(-\frac{\|P_i - P_j\|^2}{2\sigma_\gamma^2}\right) \right]. \quad (4)$$

Among them, if $x_i \neq x_j$, $\mu(x_i, x_j) = 1$, otherwise, it is 0. The rest of the expression uses two Gaussian kernels in different feature spaces. The first is the pixel position (denoted as P) and RGB color (denoted as C) between the two-way kernel. The second kernel is the pixel position. The hyperparameters ω_1, ω_2 is the weight. The hyperparameters $\sigma_\alpha, \sigma_\beta, \sigma_\gamma$ control the scale of the Gaussian kernel.

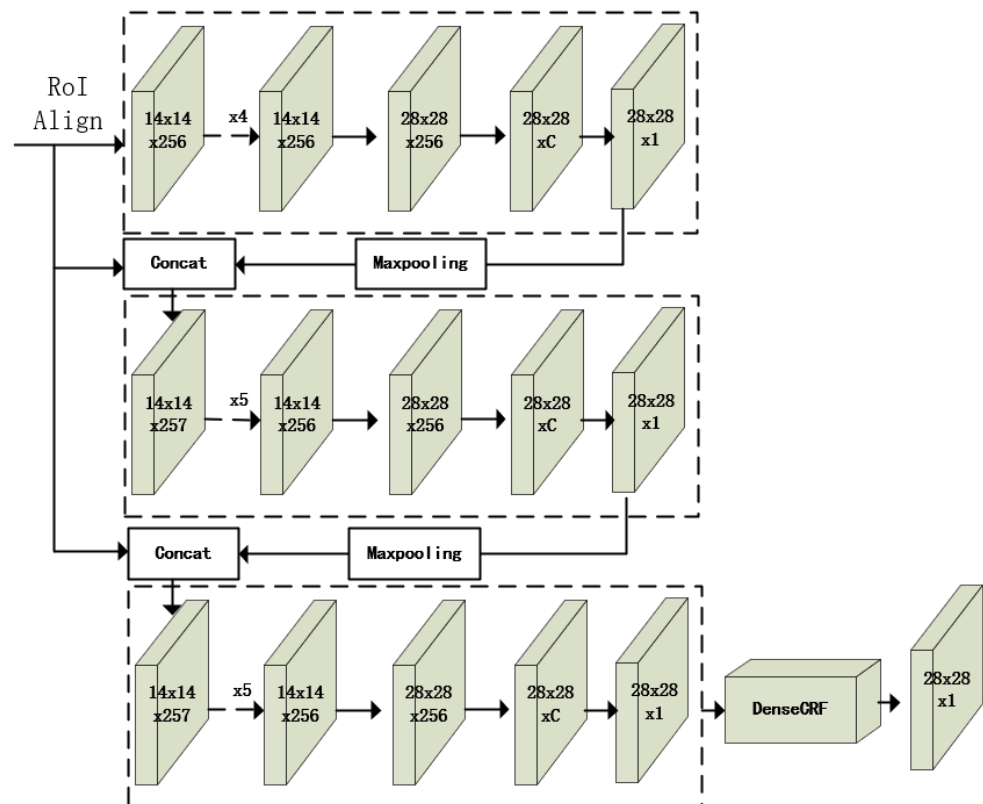


Figure 6. The structure of multiple threshold probabilistic mask branch.

The key is that this model can effectively approximate probabilistic reasoning. The full decomposition average field estimation $e(x) = \prod_i e_i(x_i)$ can be expressed as Gaussian

convolution in bilateral space. The high-dimensional filtering algorithm speeds up this calculation process, making the algorithm extremely fast in training. The unary potential energy of this model includes the shape, texture, color, and position of the image. Moreover, the binary potential energy uses contrast-sensitive, dual-core potential energy. The binary potential function of the CRF generally describes the relationship between the pixels and encourages similarity. Pixels were assigned the same label; however, pixels with larger differences were assigned different labels. The definition of this distance is related to the color value and the actual relative distance so that the CRF can segment the image at the boundary as much as possible. The difference of the fully connected CRF model is that its binary potential function describes the relationship between each pixel with all the other pixels. The model is used to establish point-to-potential energy to all pixel pairs in the image to achieve excellent refinement and segmentation.

We use an end-to-end training method to uniformly update and optimize the network parameters. The output size of the mask branch is $k \times m \times m$, and its encoding resolution is k binary masks with $m \times m$. That is, each class corresponds to one mask. The ground truth category of the RoI is k , and L_{mask} is only defined on the k -th category. That is, although k binary masks exist for each point, only one mask of k category contributes to L_{mask} . Therefore, the mask branch has no inter-class competition. The mask branch has a prediction for each category, and selects the output mask depending on the classification layer. When inferring, we still use the three-level branch because the network has a certain optimization effect on the mask, so the final mask prediction comes from the last-level mask branch.

3.4. Loss Function

The loss function in the model is used to evaluate the degree of difference between the predicted output of the model and the ground truth. In addition, it can intuitively reflect the training effectiveness of the model. Generally, a small loss value means that the predicted output is close to ground truth, and the performance of the model is better. The loss function of our model is divided into two parts. The first part is the loss function in the RPN, and the second part is the loss function of the RoI head network. The RPN is used to generate candidate bbox and fine-tune the bbox. So, our RPN loss function consists of the target recognition loss and bbox regression loss:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \tau \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*), \quad (5)$$

Where i is the index of the anchor box in the mini-batch; N_{cls} and N_{reg} are the class number of classification layers and regression layers; p_i represents the predicted probability value of the anchor as an object; $p_i^* = 1$ if the target object is in the anchor, otherwise $p_i^* = 0$; t_i represents the four parameterized coordinates of the prediction candidate box; t_i^* represents the four parameterized coordinates of the true value region; L_{cls} and L_{reg} represent the classification loss and regression loss. We add a parameter τ with a value of 10 to balance the effects of the two loss functions.

The RoI head network loss function consists of four parts: the classification loss L_{cls} , the bbox regression loss L_{box} , the segmentation loss L_{mask} , and the maskiou loss $L_{maskiou}$. The ROI head loss function of MTP R-CNN is as follows:

$$L_{head} = L_{cls} + L_{box} + L_{mask} + L_{maskiou}. \quad (6)$$

The classification loss uses the cross-entropy loss. The bbox regression loss is the smooth L1 loss function. The segmentation mask loss uses the binary cross-entropy with logits loss function, and the maskiou loss uses the mean squared error to calculate the regression loss between the predicted mask and ground truth. Therefore, the final loss function of MTP R-CNN can be expressed as follows:

$$L_{final} = L(\{p_i\}, \{t_i\}) + L_{head}. \quad (7)$$

4. Experiment

4.1. Implementation Details

To verify the effectiveness of our algorithm, all experiments were performed on the MS COCO dataset [10]. There are 80 object categories. We followed the setting of COCO 2017, using 115,000 images for training, 5000 images for validation, and 20,000 images for testing. We use the COCO evaluation index average precision (AP) to report the results. The AP includes AP@0.5 (meaning the use of the IoU threshold 0.5), AP@0.75 (the use of the IoU threshold 0.75), and the AP by size: APs (small objects with an area less than 322), APM (objects with an area between 322 and 962), and APL (objects with an area greater than 962).

We use ResNet-18/50/101 as the backbone and use the bbox branch of the Mask Scoring R-CNN as the bbox branch of the network for simulation research. In the experiment, we conducted multiple comparison experiments and ablation experiments. Except for comparative experiments, every part of the network has the same architecture. According to the experience for Cascade R-CNN, our three-level branch uses [0.4, 0.6, 0.7] as thresholds. We trained all networks for 20 epochs and reduced the learning rate by 0.1 times after 13 epochs and 18 epochs. We employed a Momentum 0.9 synchronized SGD as the optimizer. All experimental codes are written based on PyTorch for a fair comparison. The data in the experimental results are all the results of image segmentation by the network. Because our improvement does not affect bbox detection, our experiment did not involve the bbox detection result.

4.2. Multiple-Threshold Probabilistic R-CNN with Different Backbones

In this part, we tested the improvement of the network using different ResNet versions. The experiment is trained and tested on the COCO 2017 dataset. We only list the results of image segmentation. The results are listed in Table 1, showing the improvement from ResNet-50 to ResNet-101 is more obvious. In other words, better feature maps can make the segmentation work more accurate.

Table 1. Result of network verification on the COCO 2017 test dataset. AP represents the results of image segmentation.

Method	Backbone	AP	AP@0.5	AP@0.75
MTP R-CNN	ResNet-18	34.3	53.9	33.2
	ResNet-50	36.9	57.2	39.8
	ResNet-101	40.9	63.0	44.3

4.3. Ablation Experiments

To better understand the content of this article, we conducted some ablation experiments. All ablation experiments use ResNet-101+FPN. We added our improvements to the Mask Scoring R-CNN baseline and observed the results of the experiment. The results are presented in Table 2. The feature fusion design of the feature pyramid improves the network greatly, with an improvement of about 1.6 in AP, and the multiple-threshold branch also has an improvement of about 1 in AP. So, we can observe the improvement to the old network. There is about 2.6 in AP improvement. At the same time, we can observe the best performance of the network.

Table 2. Ablation experiments. We use ResNet-101+FPN as the backbone to train and test on the COCO 2017 dataset. The results without $\checkmark$ are those of the Mask Scoring R-CNN.

Method	Feature Fusion	Multiple Threshold	AP	AP@0.5	AP@0.75
baseline	$\checkmark$	$\checkmark$	38.3	58.8	41.5
			39.9	61.6	43.7
			40.9	63.0	44.3

4.4. Comparison with the State-of-the-Art

Multiple-threshold probabilistic R-CNN uses ResNet-101+FPN as the backbone and compares it with other more advanced image segmentation networks in Table 3. We show the comparison of Mask Scoring R-CNN and MTP R-CNN in Figure 7. The settings are the same as the previous ones. The data in the table are the result of the algorithm on the COCO 2017 test-dev. The training uses 280,000 iterations, and the learning rate drops at 160,000 and 240,000 iterations. The number of ROIs has also increased to 512. All the most-advanced detectors compared are trained with the IoU = 0.5. In general, our network has excellent performance compared to other mainstream networks.

Table 3. Comparison with the current more advanced network. The results are all trained and tested on the COCO 2017 dataset.

Method	Backbone	AP	AP@0.5	AP@0.75	APS	APM	APL
FCIS+++ [17]	ResNet-101	33.6	54.5	-	-	-	-
MaskLab [18]		35.5	57.4	37.4	16.9	38.2	49.2
MaskLab+ [18]		37.4	59.8	36.6	19.1	40.5	50.6
JMTC [19]	ResNet-101+FPN	37.4	60.8	39.6	18.8	40.6	53.9
SipMask [20]		38.1	60.2	40.8	17.8	40.8	54.3
Mask R-CNN [5]	ResNet-101+FPN+DCN	38.4	61.2	41.1	18.0	40.6	55.2
MS R-CNN [6]		39.6	60.7	43.1	18.8	41.5	56.2
MTP R-CNN	ResNet-101+FPN _{our}	40.9	63.0	44.3	19.2	42.0	56.5

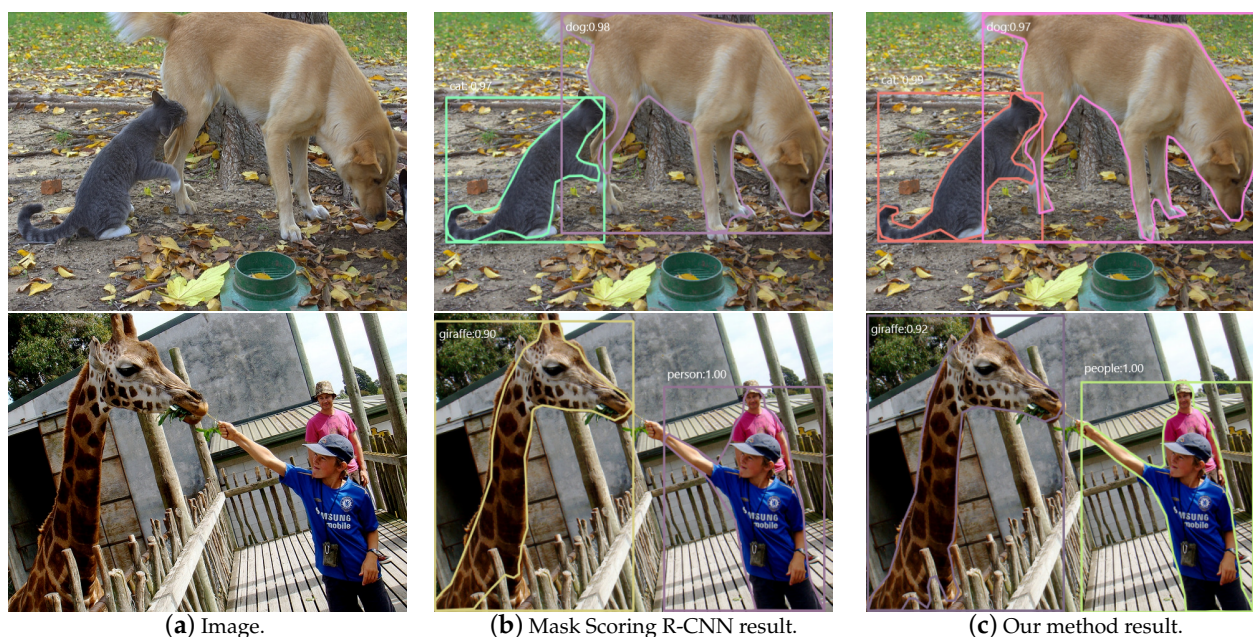


Figure 7. Comparison of Mask Scoring R-CNN and MTP R-CNN. It can be seen that our algorithm performs better in details. In particular, the processing of object edges is more accurate.

5. Conclusions

Image semantic segmentation is a challenging task. Current algorithms still have many problems; for example, the Mask Scoring R-CNN is not sensitive enough for image edge processing. Based on this, we optimize the algorithm from two aspects: feature extraction and segmentation branch. We propose a new feature fusion method to optimize the feature pyramid network. Additionally, a multiple-thresholds branch and the probability model DenseCRF are integrated into the model, which can better adapt to the segmentation of image edges, improving the segmentation effect. This method has achieved good results in different scenarios.

However, our work still has many shortcomings. Our model is complicated and the image semantic segmentation is not completely correct. In future work, we aim to study how to perform semantic segmentation on images quickly, simply and more accurately, and to determine how to segment images under weak labeled and unlabeled conditions.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; software, J.L.; validation, K.Z. and W.L.; formal analysis, K.Z.; investigation, W.L.; resources, J.L.; data curation, Y.G.; writing—original draft preparation, J.L.; writing—review and editing, Y.G. and J.Z.; visualization, J.L., K.Z. and W.L.; supervision, J.Z.; project administration, Y.G. and J.Z.; funding acquisition, Y.G. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (No. 2019YFB1404700).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The authors are very thankful to the editor and referees for their valuable comments and suggestions for improving the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
- Pinheiro, P.O.; Collobert, R.; Dollár, P. Learning to segment object candidates. *arXiv* **2015**, arXiv:1506.06204.
- He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
- Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask scoring r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 6409–6418.
- Tian, Y.; Yang, G.; Wang, Z.; Li, E.; Liang, Z. Instance segmentation of apple flowers using the improved mask R-CNN model. *Biosyst. Eng.* **2020**, *193*, 264–278. [[CrossRef](#)]
- Zhang, B.; Zhang, J. A Traffic Surveillance System for Obtaining Comprehensive Information of the Passing Vehicles Based on Instance Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2020**. [[CrossRef](#)]
- Liu, D.; Zhang, D.; Song, Y.; Huang, H.; Cai, W. Cell r-cnn v3: A novel panoptic paradigm for instance segmentation in biomedical images. *arXiv* **2020**, arXiv:2002.06345.
- Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
- Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [[CrossRef](#)] [[PubMed](#)]
- Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
- Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.

15. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Krähenbühl, P.; Koltun, V. Efficient Inference in Fully Connected Crfs with Gaussian Edge Potentials. *arXiv* **2011**, arXiv:1210.5644.
17. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2359–2367.
18. Chen, L.C.; Hermans, A.; Papandreou, G.; Schroff, F.; Wang, P.; Adam, H. Masklab: Instance segmentation by refining object detection with semantic and direction features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4013–4022.
19. Wen, Y.; Hu, F.; Ren, J.; Shang, X.; Li, L.; Xi, X. Joint multi-task cascade for instance segmentation. *J. Real-Time Image Process.* **2020**, *17*, 1983–1989. [[CrossRef](#)]
20. Cao, J.; Anwer, R.M.; Cholakal, H.; Khan, F.S.; Pang, Y.; Shao, L. SipMask: Spatial Information Preservation for Fast Image and Video Instance Segmentation. *arXiv* **2020**, arXiv:2007.14772.