Mathematical Biosciences and Engineering

MBE, 20(7): 12772–12801.
DOI: 10.3934/mbe.2023570
Received: 12 April 2023
Revised: 19 May 2023
Accepted: 24 May 2023
Published: 31 May 2023

http://www.aimspress.com/journal/MBE

Research article

Element detection and segmentation of mathematical function graphs based on improved Mask R-CNN

Jiale Lu, Jianjun Chen*, Taihua Xu, Jingjing Song and Xibei Yang

School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang 212100, Jiangsu, China

* **Correspondence:** Email: jianjunchen@just.edu.cn.

Abstract: There are approximately 2.2 billion people around the world with varying degrees of visual impairments. Among them, individuals with severe visual impairments predominantly rely on hearing and touch to gather external information. At present, there are limited reading materials for the visually impaired, mostly in the form of audio or text, which cannot satisfy the needs for the visually impaired to comprehend graphical content. Although many scholars have devoted their efforts to investigating methods for converting visual images into tactile graphics, tactile graphic translation fails to meet the reading needs of visually impaired individuals due to image type diversity and limitations in image recognition technology. The primary goal of this paper is to enable the visually impaired to gain a greater understanding of the natural sciences by transforming images of mathematical functions into an electronic format for the production of tactile graphics. In an effort to enhance the accuracy and efficiency of graph element recognition and segmentation of function graphs, this paper proposes an MA Mask R-CNN model which utilizes MA ConvNeXt as its improved feature extraction backbone network and MA BiFPN as its improved feature fusion network. The MA ConvNeXt is a novel feature extraction network proposed in this paper, while the MA BiFPN is a novel feature fusion network introduced in this paper. This model combines the information of local relations, global relations and different channels to form an attention mechanism that is able to establish multiple connections, thus increasing the detection capability of the original Mask R-CNN model on slender and multi-type targets by combining a variety of multi-scale features. Finally, the experimental results show that MA Mask R-CNN attains an 89.6% mAP value for target detection and 72.3% mAP value for target segmentation in the instance segmentation of function graphs. This results in a 9% mAP improvement for target detection and 12.8% mAP improvement for target segmentation compared to the original Mask R-CNN.

Keywords: function graphs; tactile graphics; instance segmentation; Mask R-CNN; attention mechanism

1. Introduction

A survey of the World Health Organization reveals that in 2019, there were about 2.2 billion individuals globally with visual impairments [1], and in 2020, the total number of people who were completely blind worldwide amounted to approximately 75 million. With impaired visual functions, people with severe visual impairments cannot experience the same visual pleasures as those with normal vision.

At present, people with severe visual impairments mostly depend on the auditory and tactile senses in order to gain external information. For example, audiobooks can be listened to, as can auditory readers and computer screen reading software. Meanwhile, there are texts and tactile graphics that can be felt by the fingers. These include paper braille publications and paper tactile graphics, as well as Braille e-books, Braille displays and tactile graphic displays [2,3].

A survey found that the China Braille Publishing House produces and distributes approximately 1000 titles of Braille books and periodicals each year, with a total circulation of around 320,000 copies. This averages at about 12,900 titles and 40 copies per publication, which is significantly lower than the average number of publications available to those with normal vision [4]. At present, the majority of reading materials for visually impaired individuals are either audiobooks or braille books. However, these options do not adequately satisfy the need of those with vision impairments for perception of images, which must be converted into tactile graphics in order for them to be touch-read by the blind. Tactile graphics refers to images which can be perceived through touch, composed of convex and concave lines, points and surface textures. Currently, the production of tactile graphics is not fully automated, and most of this work is done by hand, requiring a great deal of time and labor. Furthermore, the production of tactile graphics requires personnel with specialized knowledge to complete. So far, most of this work is done by teachers in schools for the visually impaired. Due to varying abilities and experiences in reading tactile maps, different blind people require tactile graphics of the same content to be presented in different forms, such as different sizes and hierarchical representations. Therefore, in order to reduce the burden on educators, enhance the efficiency of the production of tactile graphics and reduce production costs, the development of an automated tactile graphics creation method with the help of computer technology is of great significance.

The most critical aspect of creating books for the visually impaired is to make them electronic, as electronic books can be easily accessed by the visually impaired in the form of multimedia or tactile graphics. Currently, various scientists and organizations have devoted their attention to the conversion of printed books into electronic books (printed books, abbreviated as e-books). The Digital Accessible Information System (DAISY) alliance [5], a joint effort of European and American countries, aims to provide a universally accessible solution for people with Dyslexia to access digital audiobooks. The specifications necessary for creating DAISY files include 1) text files (.html or .xml) prepared in HTML (Hypertext Markup Language) or XML (Extensive Markup Language), 2) audio files (.mp3, .wav, etc.), 3) text and voice synchronization file (.smil) and 4) image files in scalable vector graphics (SVG) format written in XML [6–8]. At present, the production of digital audiobooks based on DAISY standards has been widely accepted by countries

around the world. Furthermore, the Electronic Publication (EPUB) [9] standard set by the International Digital Publishing Forum (IDPF) is compatible with DAISY standards in terms of document and image preservation. Consequently, during the electronic conversion of scientific publications, images need to be converted from the Dot Image format to SVG format. Nevertheless, at the moment, paper images can only be saved as bitmaps by scanning, and bitmaps cannot be automatically converted to SVG.

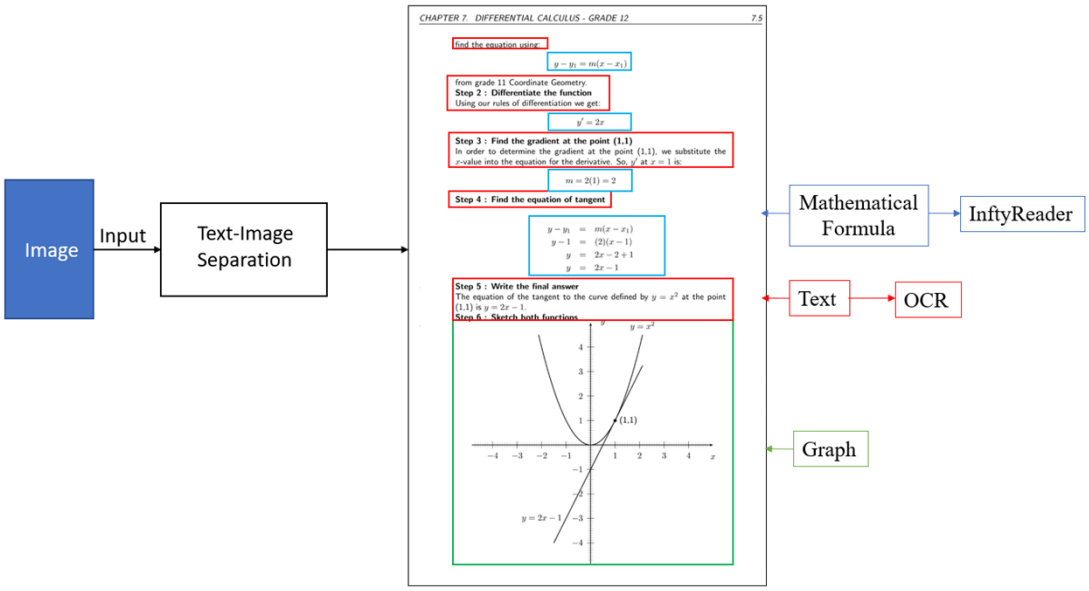


Figure 1. Basic steps of book digitization.

Taking the mathematics textbook in Figure 1 as an illustration, the basic steps of book digitization are as follows: First, the separation of text content (including mathematical expressions) and graphs can be accomplished using the text-graph separation technology proposed by P. P. Rege et al. [10], the end-to-end trainable neural network proposed by P. Lyu et al. [11] or the character region awareness technology proposed by Y. Baek et al. [12]. Second, the recognition and digitization of text content can be achieved through the use of optical character recognition (OCR) [13] technology. The development of OCR technology has reached a high level of maturity, and existing technologies such as Tesseract OCR [14] and East [15] make this task easier. The mathematical expressions within the text can be recognized and digitized with InftyReader [16], which is developed by the InftyProject [17]. In addition, deep neural networks with sequence-level training proposed by Z. Wang et al. [18] can also be utilized for this purpose. Both of these techniques can translate the original mathematical expressions in the textual content of books into formats such as LaTeX and MathML. Lastly, the segregated graphs need to be identified and digitized. Currently, although the ImageCaption [19] method such as the end-to-end transformer based model proposed by Y. Wang et al. [20] can achieve semantic description of images, it is primarily applied for describing daily actions. Upon examining Figure 1, it is evident that the function graphs comprise coordinate axes and quadratic functions. Currently, there is no available technology to recognize and convert the graph elements into SVG format for digitizing function graphs.

During the electronic conversion of graphs, there are a series of research methods [21–24] that

can digitize function graphs; however, these methods require that the graphs meet certain criteria. For instance, the graphs must be confined to a specific area (delineated by the x and y axes). A. Balaji et al. [25] analyzed the labeled area through the connected component and fitted the labeled area through the minimum surrounding rectangle, but they only managed to extract the bar graph in the chart. Some scholars, such as J. Chen et al. [26], recognize and redraw function graphs using pattern recognition and save them as SVG. However, the recognition accuracy is low, and it does not have real-time processing capabilities. On one hand, graph acquisition generally introduces noise; on the other, the feature learning abilities of the respective methods may not be sufficient. When compared to traditional graph recognition methods, deep learning models are able to automatically learn the features of the training data, and they possess stronger feature learning abilities, which in turn results in a better performance when using deep learning models for graph recognition. J. Staker et al. [27] successfully applied the deep learning method to recognize the visual representation of chemical molecular structures. Likewise, M. Oldenhof et al. [28] were also able to identify optical patterns of compounds using a deep learning model.

Many computer vision tasks today incorporate deep learning techniques. Deep learning has found widespread application across industries, leveraging its high adaptability, powerful feature learning capabilities, ability to handle large-scale data and efficiency. Thus, building upon these strengths, this paper takes a deep learning approach to tackle the instance segmentation of function graphs. In terms of feature extraction in deep learning, the Transformer network proposed by A. Vaswani et al. [29] quickly dominated the field of natural language processing as soon as it was introduced. With subsequent applications of Transformer and its improved models such as ViT proposed by A. Dosovitskiy et al. [30] and Swin Transformer proposed by Z. Liu et al. [31] in the computer vision domain, networks with Transformer as the backbone have emerged as popular directions, surpassing traditional convolution-based networks in various tasks and achieving state-of-the-art rankings. However, the emergence of ConvNeXt has shifted scholars' attention back to convolutional networks. Z. Liu et al. [32] demonstrated through experiments that ConvNeXt outperforms Swin Transformer and achieves state-of-the-art performance in various image processing tasks with lower computational requirements. Regarding function graphs, compared to Transformers, ConvNeXt leverages the local connectivity and weight sharing properties of convolutional operations, enabling better capture of spatial information and local features, thereby enhancing feature representation capabilities. In terms of feature fusion in deep learning, BiFPN [33] employs a simple and efficient network structure that addresses the limitations of information propagation and loss in traditional fusion networks while maintaining low computational and storage costs. For function graphs, a multi-scale feature fusion approach in BiFPN offers translational and scale invariance, improving robustness and generalization to object deformations of the model, occlusions and scale variations. Based on the analysis above, this paper explores improvements using ConvNeXt and BiFPN as the base models.

This paper proposes an improved instance segmentation model based on Mask R-CNN [34], which leverages the benefits of convolution and self-attention and effectively improves the accuracy of the model in detecting and segmenting graph elements in function graphs. The contributions of this paper are summarized as follows:

1) For feature extraction, ConvNeXt is used as the base model. By adding an attention module made up of local detail features, remote information features and channel information features to the ConvNeXt block module, the improved ConvNeXt is called MA ConvNeXt in this paper. MA

ConvNeXt is able to amplify its ability to extract local information, global information and varied channel feature information.

2) For feature fusion, this paper proposes a new model by adding an Atrous Spatial Pyramid Pooling (ASPP) [35] module and Residual Feature Augmentation (RFA) [36] module to BiFPN, which is called MA BiFPN. The proposed model can enhance the detection capability at different scales.

3) In order to improve the segmentation capability of the model, the aspect ratio of the proposed candidate anchor boxes in the Region Proposal Network (RPN) layer of the model is modified, and a penalty factor is included in the calculation of the Intersection-over-Union (IoU) metric in order to create a margin space in the target detection box.

4) The MA Mask R-CNN model is proposed to detect and segment 13 kinds of subdivided mathematical function graphs, reaching a detection accuracy of 89.6% and mask segmentation mAP score of 72.3%. Through multiple comparison tests, it has been proven that MA Mask R-CNN performs better in terms of category average accuracy and mask segmentation quality of function graphs.

The remaining organization of this paper is as follows. Section 2 introduces the specific structure of the original Mask R-CNN model. Section 3 describes the improved network structure proposed in this paper. It also details the specific improvements and loss functions of this network in comparison to the original Mask R-CNN model. These improvements include feature extraction, feature fusion, region proposal network and mask segmentation methods. Section 4 reports on the datasets used, as well as the comparative experimental results and corresponding analyses of each model on the dataset. Finally, Section 5 summarizes the work of this paper and outlines future directions and key points.

2. Original model

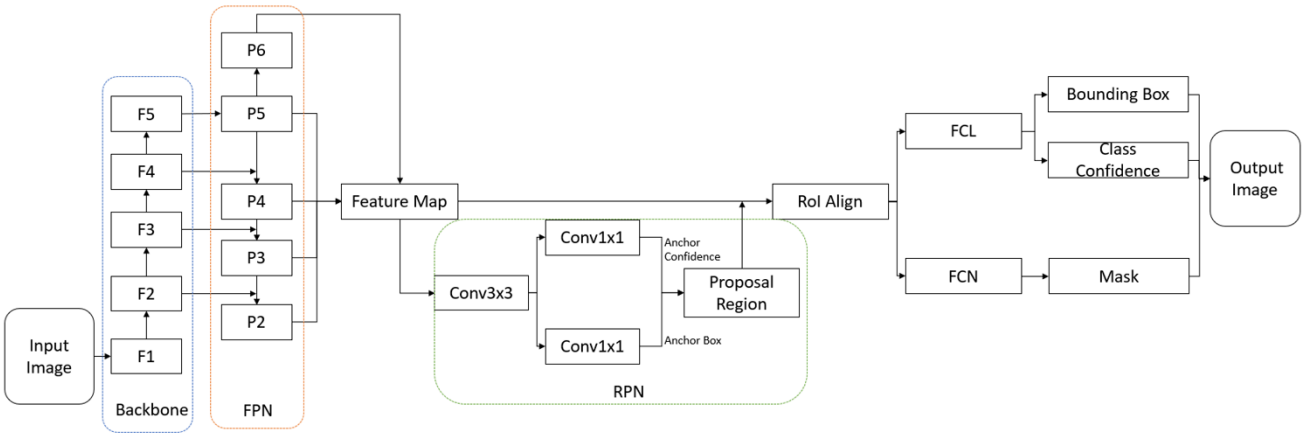


Figure 2. The original Mask R-CNN model.

Mask R-CNN is a model, after adding branches for predicting target segmentation mask, based on Faster R-CNN [37]. Since it was proposed, it has achieved good results in object detection and segmentation. It is a classic model in the instance segmentation task. Figure 2 shows the overall framework of Mask R-CNN. The steps are as follows: 1) The original image is input to Residual Network (ResNet) [38], which is a feature extraction network to generate feature maps. 2) The feature maps of the four stages in the feature extraction network are input to the Feature Pyramid

Networks (FPN) [39], which are feature fusion networks to obtain new feature maps that combine high-level feature maps with low-level feature maps. 3) Input the feature maps generated by FPN into RPN and RoI Align. Anchor boxes and anchor confidence of feature maps are obtained in RPN, and Non-Maximum Suppression (NMS) is used to eliminate anchor boxes with low confidence. Since anchor sizes generated in RPN are inconsistent, different sizes of anchor are mapped into fixed area sizes in RoI Align. 4) After the above steps, the feature maps fixed by RoI Align are input into the Fully Convolutional Layer (FCL) and Fully Convolutional Network (FCN). Bounding box and class confidence of the target instance are output after FCL, and mask of the target instance is output after FCN.

3. Improved model

3.1. Overall architecture of MA Mask R-CNN

MA Mask R-CNN is an improved model based on Mask R-CNN. In MA Mask R-CNN, ResNet feature extraction network of Mask R-CNN is replaced by MA ConvNeXt network, and MA BiFPN is used to replace the original FPN in feature fusion network. The mask segmentation strategy is replaced by the original FCN with PointRend [40]. The frame diagram of the overall model is shown in Figure 3.

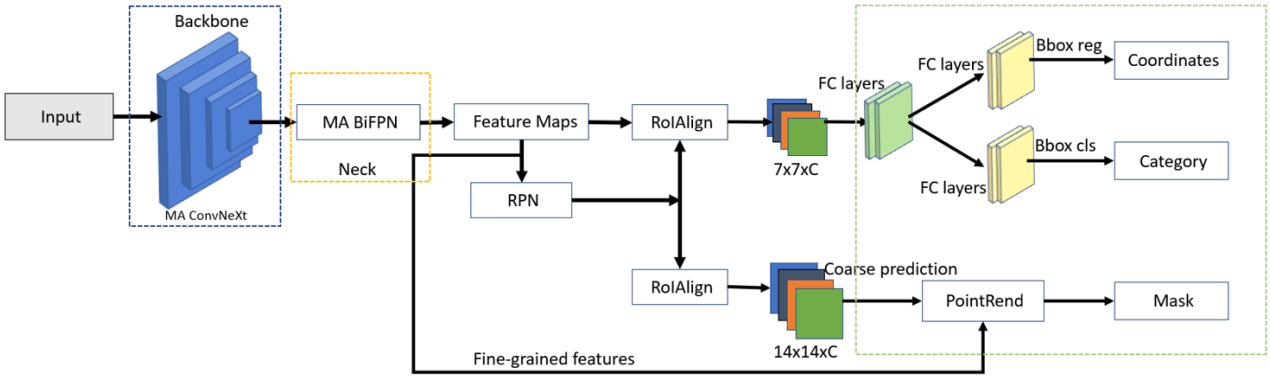


Figure 3. The overall framework of the MA Mask R-CNN model for graph detection and segmentation of mathematical functions proposed in this study.

The process of the MA Mask R-CNN model is as follows: 1) Input mathematical function graph into the model, enter the feature extraction network, which is MA ConvNeXt, and generate corresponding feature maps according to the four feature extraction stages in MA ConvNeXt. Compared with the original model ConvNeXt, MA ConvNeXt added an additional attention module with local, global and channel information at the same time, so that the generated feature maps have more advanced semantic information and positioning capabilities. 2) Input the feature maps generated by MA ConvNeXt into MA BiFPN, which is a feature fusion network. In order to enhance the feature fusion network's recognition ability in slender targets and multi-scale information fusion ability, an RFA module is added to the original BiFPN after the top-level semantic information of the top-to-bottom branch. ASPP modules are added to the input feature maps of the top-to-bottom and bottom-to-top branches. 3) Adjust RPN. In RPN, in order to make the detection box not completely

close to the target, appropriately increase the size proportion of the anchor box, and add a penalty factor to IoU to obtain a more suitable detection box and enhance the segmentation effect. The foreground and background are classified according to the adjusted RPN, and a regression operation is carried out on the bounding box. After the NMS removes the Anchor box with low confidence, the proposed region is finally generated. 4) Put the proposed region into ROI to output the feature map of the same size, and conduct the regression classification of the graph element, so as to generate the final prediction box and segmentation mask. In the mask branch, PointRend is used instead of FCN to obtain a smoother and finer pixel mask.

3.2. MA ConvNeXt network for feature extraction

In the aspect of a feature extraction network, an improved network named MA ConvNeXt based on ConvNeXt is proposed in this study. The overall network structure of MA ConvNeXt is shown in Figure 4. The ConvNeXt-T network is selected as the foundation network. Compared to Transformer based networks, ConvNeXt does not require complex operations such as block merging, sliding windows and relative location indexing, thus providing superior performance and less computation. ConvNeXt uses Transformer network for reference in network structure design, and it uses the methods in convolution network for replacement and optimization. The components of the MA ConvNeXt network constructed in this study are mainly the convolution layer, which is called MA ConvNeXt Block. MA ConvNeXt Block is the core part of feature extraction network, which is responsible for extracting features of mathematical function graphs. In order to deepen the network to obtain better and more features, the stacking times of the four MA ConvNeXt Block layers are set to (3, 3, 9, 3) according to the original ConvNeXt-T network structure. In MA ConvNeXt, first, a 4 × 4 convolution kernel with step size set to 4 is used for the convolution operation, and then layer normalization is carried out to initially refine features and improve the distribution of features in feature map. After the above processing, feature maps are input into the MA ConvNeXt Block for attentional multi-scale feature extraction.

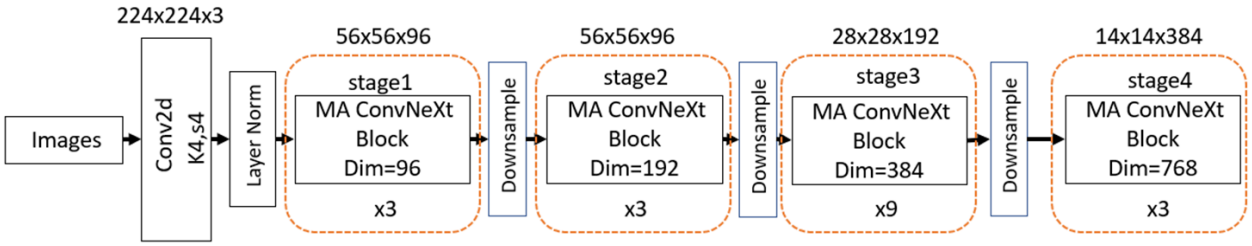
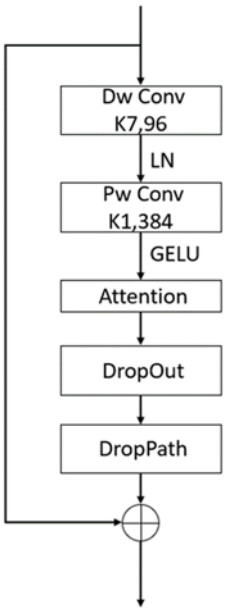
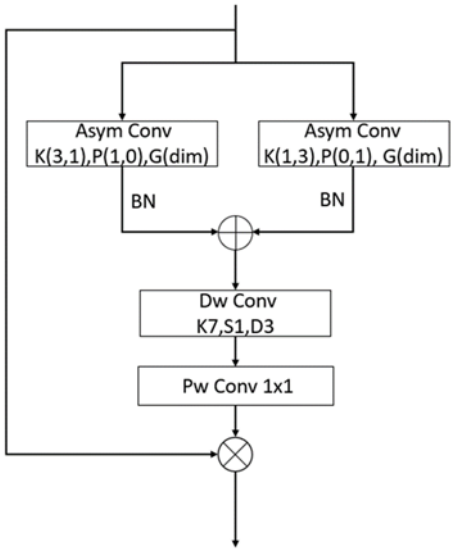


Figure 4. MA ConvNeXt feature extraction overall network structure.

The general structure of the MA ConvNeXt Block is shown in Figure 5(a). In MA ConvNeXt Block, first, a Dw Conv is carried out on the features. In order to make effective use of the information of different layers in the same spatial position, the processed features continue to be input to Pw Conv after layer normalization (LN), which generates a new feature map by weighted combination of the input independent feature maps in the depth direction. The above methods greatly reduce the amount of computation and parameters and can make the network deeper with the same parameters. The feature maps are then input into the Attention module by introducing nonlinear properties through activation functions.



(a) Structure of MA ConvNeXt Block.



(b) Attention module; The Attention module in (a), where K is the size of the convolution kernel, P is the padding, and G is the grouping. Dw Conv is depthwise convolution, and Pw Conv is pointwise convolution. Asym Conv is asymmetric convolution.

Figure 5. MA ConvNeXt Block structure drawing and detailed Attention structure.

In this study, it is hoped that the Attention module is able to get local attention as well as good remote information. In general, local attention can be completed by small convolution kernels, while remote information can only be completed by large convolution kernels except for the self-attention mechanism. However, the self-attention mechanism destroys the two-dimensional property of graphs, and large convolution kernels require a large amount of computation. In order to avoid the respective

shortcomings of the self-attention mechanism and the large nuclear convolution, the Attention module in the MA ConvNeXt Block is shown in Figure 5(b). The Attention module can be expressed by Eqs (1) and (2).

$$Attention = Conv_{1\times1}\left(\text{Dw Conv}\left(Asym\ Conv_{3\times1}(F) \oplus Asym\ Conv_{1\times3}(F)\right)\right) \tag{1}$$

$$Output = Attention \otimes F \tag{2}$$

where $Attention \in \mathbb{R}^{C\times H\times W}$ denotes attention, and its value denotes the importance of different features. $F \in \mathbb{R}^{C\times H\times W}$ denotes input feature maps. $Asym\ Conv_{3\times1}$ denotes asymmetric convolution with a convolution kernel of size 3×1. $Asym\ Conv_{1\times3}$ denotes asymmetric convolution with a convolution kernel of size 1×3. $\oplus$ denotes element-wise addition. $\otimes$ denotes an element-wise product.

The operation steps of the Attention module are as follows: First, the features are input into the asymmetric convolution [41]. An ordinary 3×3 convolution is replaced by an asymmetric convolution of feature maps whose kernels are (3,1) and (1,3), respectively. This method reduces the amount of computation and can not only enhance the feature extraction of slender objects such as mathematical function graphs but also make up for the information loss caused by the following Dw Conv dilation rate of 3. Second, separate BN operations are carried out before the fusion of two asymmetric convolution branch feature maps to avoid unifying BN and weakening the feature maps between different convolutions. Then, by using the convolution of 7×7 large kernel Dw Conv with a dilatation rate of 3 and a 1×1 Pw Conv, the fused features after BN operation can obtain better remote information and receptive field. Finally, the result of the previous step is used as the weight of the original different feature maps, so as to play the role of attention.

Algorithm 1: MA ConvNeXt

Input: Input feature X.
Output: Output feature F_i.
Initialize $In\_channel = 3$;
Initialize Stacking Counts $N = [3,3,9,3]$, $Dims = [96,192,384,768]$;
$X = Conv(In\_channel, Dims[0], Kernel\_size = 4, Stride = 4)(X)$;
$X = LayerNorm(Dims[0])(X)$;
for $i = 0$ to $len(N) - 1$ **do**
  for $j = 0$ to $N[i]$ **do**
    Set $X' = X$
    Set $X = DwConv(Dims[i], Kernel_{size} = 7, Padding = 3)(X)$;
    Set $X = LayerNorm(Dims[i])(X)$;
    Set $X = PwConv(Dims[i], 4 * Dims[i])(X)$;
    Set $X = Attention\big(GELU(X)\big)$;
    Set $X = DropPath\big(DropOut(X)\big)$;
    Set $X = X' + X$;
  End
  Set $F_i = X$;
  Return F_i.
  Set $X = Downsample(F_i)$;
End

The input features pass through the Attention module and output the features with attention into the final part of the MA ConvNeXt Block. In order to prevent model overfitting and improve generalization ability, DropOut and DropPath layers are added at the end of MA ConvNeXt Block. Both work similarly. DropOut deactivates neurons at a certain rate, while DropPath deactivates the master branch structure at a certain rate. To integrate with the preceding content and provide a comprehensive and intuitive presentation of MA ConvNeXt, Algorithm 1 also provides the pseudocode of MA ConvNeXt.

3.3. MA BiFPN network for feature fusion

The MA BiFPN feature fusion network is proposed based on the BiFPN network. Mask R-CNN uses FPN as a feature fusion network. FPN has the following defects in feature information fusion: 1) There is information loss in the process of adjacent scale feature fusion: semantic information loss caused by fewer channels in the process of fusion from high-level to low-level. 2) Because of the top-to-bottom or bottom-to-top structure of FPN, it pays more attention to the feature maps of adjacent layers, so the feature maps of the high-level layer cannot be directly transferred to the lower layer but must pass through the layers, which will lead to the loss of high-level semantic information. 3) The characteristic information of each scale is inconsistent.

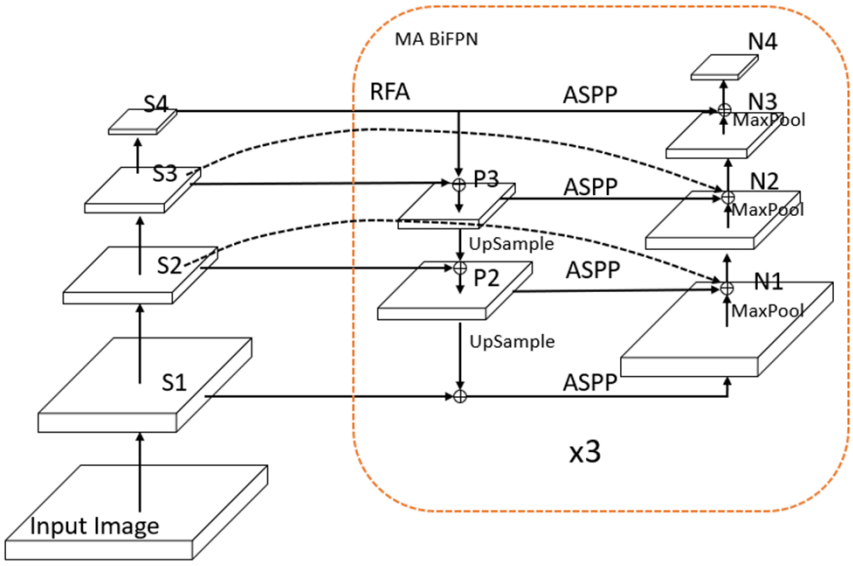


Figure 6. Structure of feature fusion network MA BiFPN.

BiFPN builds a bottom-to-top path based on FPN to better retain the low-level feature information and adds extra weight to each input to learn the importance of each feature. BiFPN goes on to add residual joins, remove nodes with only one input edge and do weight fusion and iterate the entire fusion network three times. BiFPN increases the residual connection to enhance the representation of features. BiFPN removing the node with a single input edge is because the node with single input variable does not have feature fusion and has less information and low contribution. BiFPN adds a weight for each scale feature map to reflect the scale contribution. Equation (3) shows

the weighted fusion calculation method.

$$Output = \sum_i \frac{W_i}{e + \sum_j W_j} \cdot Input_i \tag{3}$$

where W_i is the learnable weight, e is the set smaller learning rate, and the final $Output$ value is between 0 and 1.

The BiFPN has the above improvements, but it still has not solved the problems of poor semantic information fusion effect in non-adjacent layers and information loss caused by multi-scale information transmission. Therefore, this study proposes an improved feature fusion network, which is called MA BiFPN, and the specific structure is shown in Figure 6, where S_1, S_2, S_3 and S_4 are output features of the four stages in MA ConvNeXt, respectively.

In order to avoid the information loss caused by the reduction of the number of channels when the highest-level feature map is transmitted to the feature fusion network and to strengthen the contextual information compatible with other layers, this study replaces the path of passing the S_4 feature map of the highest-level feature extraction network to P_3 and N_4 with the RFA module.

The structure of the RFA module is shown in Figure 7. Using ratio-invariant adaptive pooling on the S_4 layer generates multi-scale feature maps. A 1×1 convolution is used to generate the output feature with a channel number of 256, which is then up-sampled to the same scale by bilinear interpolation. The Adaptive Spatial Fusion module is used to adaptively combine these contextual features to reduce the blurring effect caused by the interpolation. The Adaptive Spatial Fusion module assigns weights to each feature map and then aggregates these contextual features into a new feature map, as shown in Eq (4) for the weighted feature fusion computation of the Adaptive Spatial Fusion module.

$$O_{ij}^l = \alpha_{ij}^l \cdot in_{ij}^{1\to l} + \beta_{ij}^l \cdot in_{ij}^{2\to l} + \gamma_{ij}^l \cdot in_{ij}^{3\to l} \tag{4}$$

where O_{ij}^l denotes the feature vector at (i, j) of the l-th layer. α_{ij}^l, β_{ij}^l, γ_{ij}^l denote the remaining three layers' learnable weights (i.e., contribution degree) on the feature maps. $in_{ij}^{n\to l}$ denotes the feature vector generated at (i, j) of the l-th layer after the feature map is resized from the n-th layer. The newly generated feature map is then combined with the S_3 feature map in the following feature fusion network and propagated to the lower-level feature maps for fusion, and it is then processed by the ASPP module and combined with the N_3 feature map to generate the N_4 feature map. Equations (5) and (6) illustrate the feature fusion process of the improved MA BiFPN at layer N_3.

$$P_3 = Conv\left(\frac{W_1 \cdot S_3 + W_2 \cdot Resize\big(RFA(S_4)\big)}{W_1 + W_2 + e}\right) \tag{5}$$

$$N_3^{out} = Conv\left(\frac{W_1' \cdot S_3 + W_2' \cdot ASPP(P_3) + W_3' \cdot Resize(N_2^{out})}{W_1' + W_2' + W_3' + e}\right) \tag{6}$$

where W_i and W_i' are all learnable parameters, and e is the learning rate. $Resize\big(RFA(S_4)\big)$ indicates that after processing by the RFA module, S_4 is upsampled and scaled to the same size as P_3. P_3 denotes the intermediate feature map at level 3 on the top-to-bottom path, and $Resize(N_2^{out})$ indicates the scaling of the output feature of N_2 to the same size as N_3.

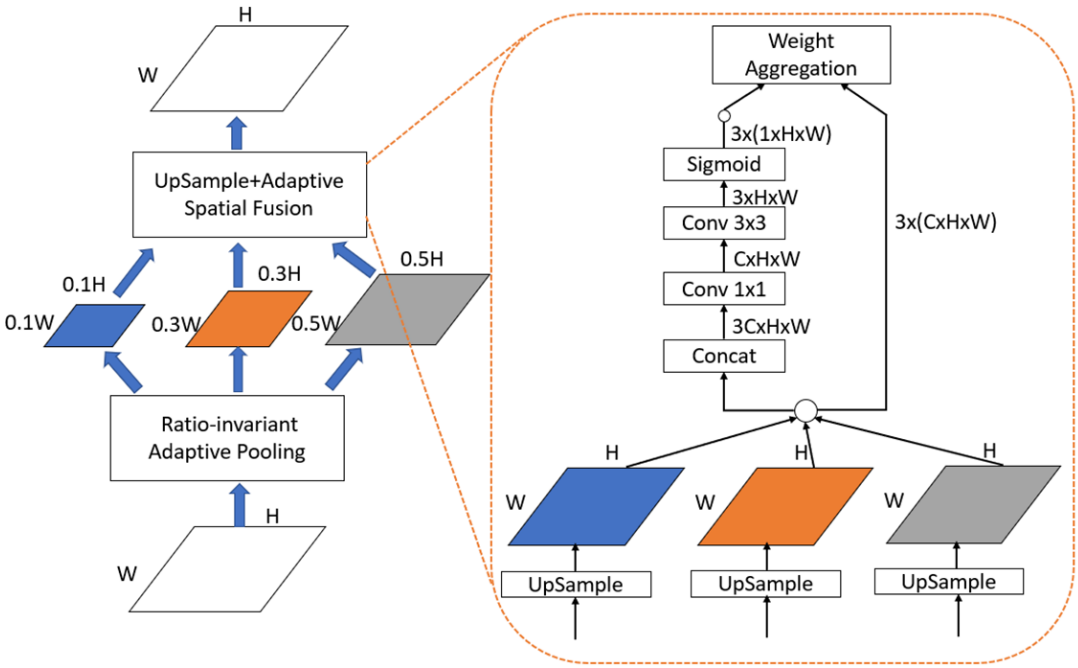


Figure 7. Structure of RFA module.

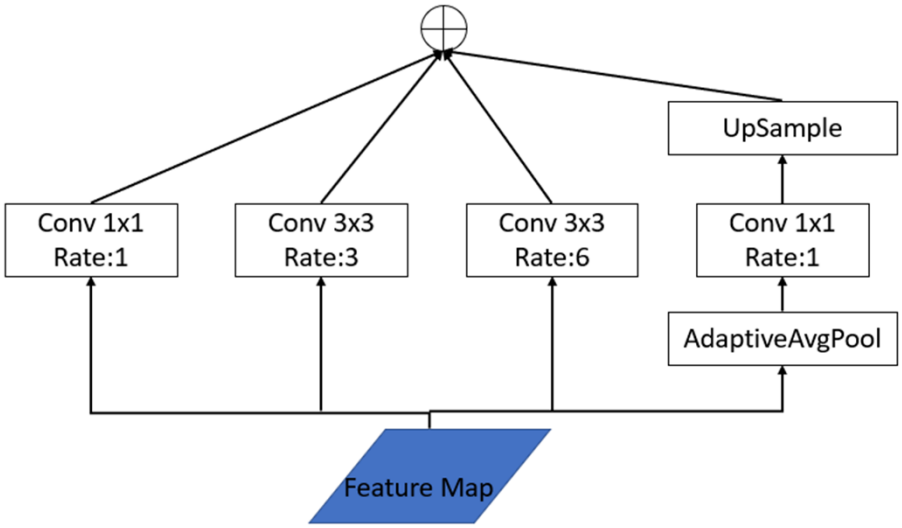


Figure 8. Structure of ASPP.

Continuing from the previous part, the feature N_4 is obtained by weighting the pooled feature N_3 and the feature S_4 after passing through the RFA and ASPP modules. Equations (7)–(10) present the feature fusion process of the highest-level feature N4 in the improved MA BiFPN.

$$R_{1,2,3} = RoiAdaptivePool(S_4) \tag{7}$$

$$Z = ASF\left(UpSample\left(R_{1,2,3}\right)\right) \tag{8}$$

$$Z_2 = \frac{1}{4}\left(\sum_{m=1}^{4} f_m(Z)\right) \tag{9}$$

$$N_4 = Conv\left(\frac{w_1 \cdot Z_2 + w_2 \cdot Resize(N_3)}{w_1 + w_2 + e}\right) \tag{10}$$

where $RoiAdaptivePool$ denotes Ratio-invariant Adaptive Pooling, $R_{1,2,3}$ denotes the three multi-scale features generated by applying Ratio-invariant Adaptive Pooling to S_4, $UpSample$ denotes the bilinear interpolation method, ASF denotes the Adaptive Spatial Fusion module, Z denotes the weighted feature obtained by ASF, m denotes the number of convolutions with different parameters in the ASPP module, f_m denotes the convolution operation with different parameter settings in the ASPP module, Z_2 denotes the feature obtained by applying the ASPP module to Z, w_i are learnable parameters, e is the learning rate, and $Resize(N_3)$ indicates the down-sampling of feature N_3 to the same scale as feature S_4.

When obtaining and extracting features of each layer, FPN only uses convolution whose convolution kernel is 1×1 to reduce the number of channels, greatly reducing the feature content of each scale. Therefore, in this study, ASPP is replaced with the path of feature transfer at each layer of feature fusion. In the BiFPN of three iterations, multi-scale feature maps are passed through ASPP to the next bottom-to-top or top-to-bottom fusion path. The concrete structure of ASPP is shown in Figure 8, where Rate stands for dilation rate. The input feature map is obtained by 4 parallel convolution branches with different dilation rates, and the multi-scale feature maps generated by the input feature map are merged at the end. ASPP can enhance the receptive field of feature fusion and enhance the extraction ability of the overall information, which is conducive to improving the final feature maps obtained by each branch of the feature fusion network. To integrate with the preceding content and provide a comprehensive and intuitive presentation of MA BiFPN, Algorithm 2 also provides the pseudocode of MA BiFPN.

Algorithm 2: MA BiFPN

Input: Feature $S_1,\ S_2,\ S_3,\ S_4$;
Output: Fused Feature $N_1,\ N_2,\ N_3,\ N_4$;
Initialize $Iteration = 3$;
Initialize $level4 = S_4,\ level3 = S_3,\ level2 = S_2,\ level1 = S_1$;
for $i = 0$ to $Iteration - 1$ **do**
    Set $T = RFA(level4)$;
    Set $P_3 = WeightedSum(T, S_3)$;
    Set $P_2 = WeightedSum(UpSample(P_3), S_2)$;
    Set $D = WeightedSum(UpSample(P_2), S_1)$;
    Set $N_1 = ASPP(D)$;
    Set $N_2 = WeightedSum(MaxPool(N_1), ASPP(P_2), level2)$;
    Set $N_3 = WeightedSum(MaxPool(N_2), ASPP(P_3), level3)$;
    Set $N_4 = WeightedSum\big(MaxPool(N_3), ASPP(T)\big)$;
    Set $level4 = N_4,\ level3 = N_3,\ level2 = N_2,\ level1 = N_1$;
End
Return $N_1,\ N_2,\ N_3,\ N_4$.

3.4. Improved region proposal network

After the MA BiFPN feature fusion network, four multi-scale fusion feature maps are obtained. These are then fed into the RPN for bounding box regression. The bounding box regression of the RPN is a rough result, and it is easy to lose information for the segmentation task. Common RPN judges candidate boxes by the IoU values of bounding boxes and candidate boxes (in target detection, it is the IoU of prediction box and ground truth box; and in segmentation, it is the IoU of prediction mask and ground truth mask). The IoU is calculated as Eq (11).

$$IoU = \frac{B_1 \cap G_2}{B_1 \cup G_2} \tag{11}$$

where B_1 is the candidate box, and G_2 is the ground truth box.

As can be seen from the equation, if the candidate box is too large, the IoU value will be too small and will be discarded. However, the candidate box which is larger than the ground truth box can contain the segmented object more comprehensively and keep more background area to reduce the information loss which may be discarded by the candidate box which is close to the edge. Therefore, in the model of this study, the selection of anchor candidate box increases by about 5%, and the penalty factor $\lambda(G_2 - B_1 \cap G_2)$ is added for IoU to punish the candidate box for not covering the target region. The improved IoU calculation equation is shown in (12).

$$IoU = \frac{B_1 \cap G_2}{B_1 \cup G_2 + \lambda(G_2 - B_1 \cap G_2)} \tag{12}$$

where λ denotes the learned penalty intensity parameter, and $(G_2 - B_1 \cap G_2)$ denotes the spatial gap between the candidate box and the ground truth box.

The improved RPN retains a certain amount of elastic space for the selected candidate boxes, providing better proposal regions for subsequent segmentation.

3.5. Mask segmentation method

Figure 9 shows the segmentation strategy of the Mask branch selected by MA Mask R-CNN. MA Mask R-CNN uses PointRend to replace the FCN in traditional Mask R-CNN for mask segmentation. PointRend innovatively introduces the rendering idea of computer graphics and regards the graph as the discretization expression of the real target. Therefore, the segmentation problem can be regarded as the prediction of the area occupied by a real target in the discretized graph. Compared with the prediction after sampling to the same size as the input graph on the FCN, PointRend is predicted directly after multiple pooled downsampling. PointRend consists of three main components: Point Selection, Point-wise and Point head. Point Selection: Among rough segmentation feature maps, difficult points with poor segmentation effect are selected; Point-wise: By combining Fine-grained features and Coarse features, point-wise feature maps are constructed in selected difficult points. Point head: A simple MLP network is trained to perform point-by-point segmentation prediction for selected difficult points' features, and the re-predicted results are replaced with the original coarse predicted results. PointRend uses the above method to reduce the information loss caused by continuous upsampling, and because it is more aimed at the goal of fewer difficult points than the whole mask pixel points, it can not only achieve accurate segmentation and smooth segmentation edge but also greatly save the consumption of computing resources.

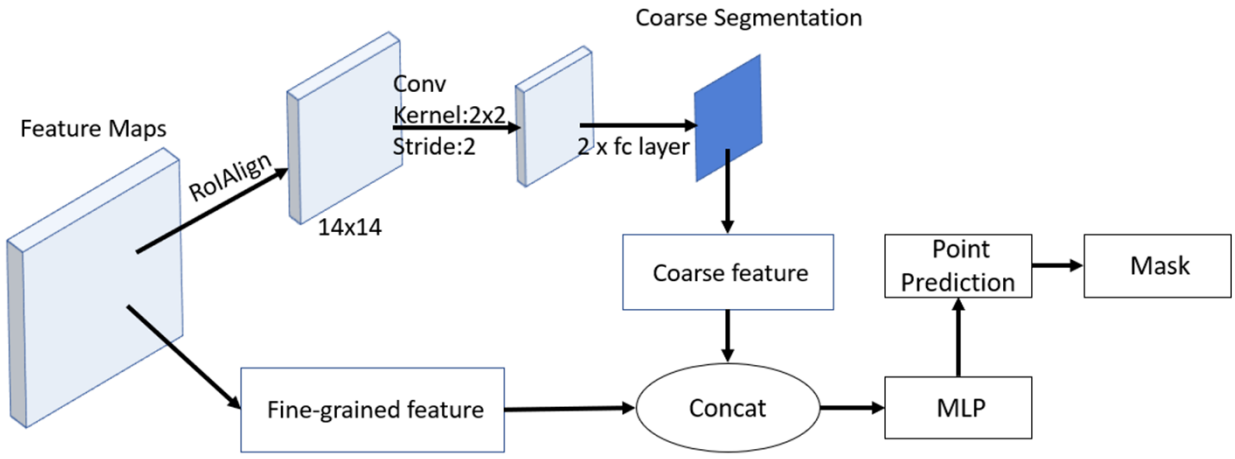


Figure 9. The segmentation strategy used by the mask branch.

3.6. Loss function

In this study, loss function L can be expressed as the sum of loss L_{rpn} at RPN stage, the final classification loss of the model, bounding box regression loss L_f and mask branch loss L_{mask} loss, as shown in Eq (13).

$$L = L_{rpn} + L_f + L_{mask} \tag{13}$$

In Eq (13), L_{rpn} is composed of classification loss and bounding box regression loss at RPN stage, as shown in Eqs (14)–(17).

$$L(\{p_i\},\{r_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(r_i, r_i^*) \tag{14}$$

$$L_{cls}(p_i, p_i^*) = -log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \tag{15}$$

$$L_{reg}(r_i, r_i^*) = \sum_i smooth_{L1}(r_i - r_i^*) \tag{16}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{17}$$

where p_i denotes the probability that the i-th anchor is predicted to be a true label, and the value of p_i^* denotes the current positive and negative samples, which is 1 when the calculated samples are positive and 0 when the calculated samples are negative. r_i denotes the bounding box regression parameter predicting the i-th anchor, and r_i^* denotes the ground truth corresponding to the i-th anchor. N_{cls} indicates the number of all samples in a batch, and N_{reg} indicates the number of anchor locations. λ is the balance parameter set. $\sum_i L_{cls}(p_i, p_i^*)$ denotes classification loss, and $\sum_i p_i^* L_{reg}(r_i, r_i^*)$ denotes bounding box return loss. L_{cls} uses Softmax Cross Entropy loss function and L_{reg} uses $smooth_{L1}$ loss function.

In Eq (13), L_f is composed of classification loss and bounding box regression loss in the final prediction stage of the model, as shown in Eq (18).

$$L(p, g, r^g, v) = L_{cls'}(p, g) + \lambda[g \geq 1]L_{loc}(r^g, v) \tag{18}$$

where p denotes the softmax probability distribution predicted by classifier $p = (p_0, \ldots, p_k)$. g denotes the real category label of the corresponding target, and r^g denotes the regression parameter of the corresponding category g predicted by the corresponding bounding box regressor $\left(r_x^g, r_y^g, r_w^g, r_h^g\right)$. v denotes the bounding box regression parameter corresponding to the real target $\left(v_x, v_y, v_w, v_h\right)$. $L_{cls'}$ still uses Softmax Cross Entropy loss function, while L_{loc} uses $smooth_{L1}$ loss function. $L_{cls'}(p, g)$ denotes classification loss, and $\lambda[g \geq 1]L_{loc}(r^g, v)$ denotes bounding box regression loss.

L_{mask} in Eq (13) adopts the Binary Cross Entropy loss function on the Mask branch:

$$L_{mask} = \frac{1}{m^2}\sum_{i}^{K}(1^k)\sum_{1}^{m^2}\left[-y * log\left(sigmoid(x)\right) - (1-y) * log\left(1 - sigmoid(x)\right)\right] \tag{19}$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{20}$$

where 1^k means 1 when the k-th channel corresponds to the true category of the target, and 0 otherwise. x denotes the output value at the current position, and $sigmoid(x)$ denotes the result of transforming the output x through a sigmoid function. y indicates the label value of the mask at the current location, which is 0 or 1. m denotes the size of the feature map.

4. Experiments

4.1. Experimental data and experimental environment

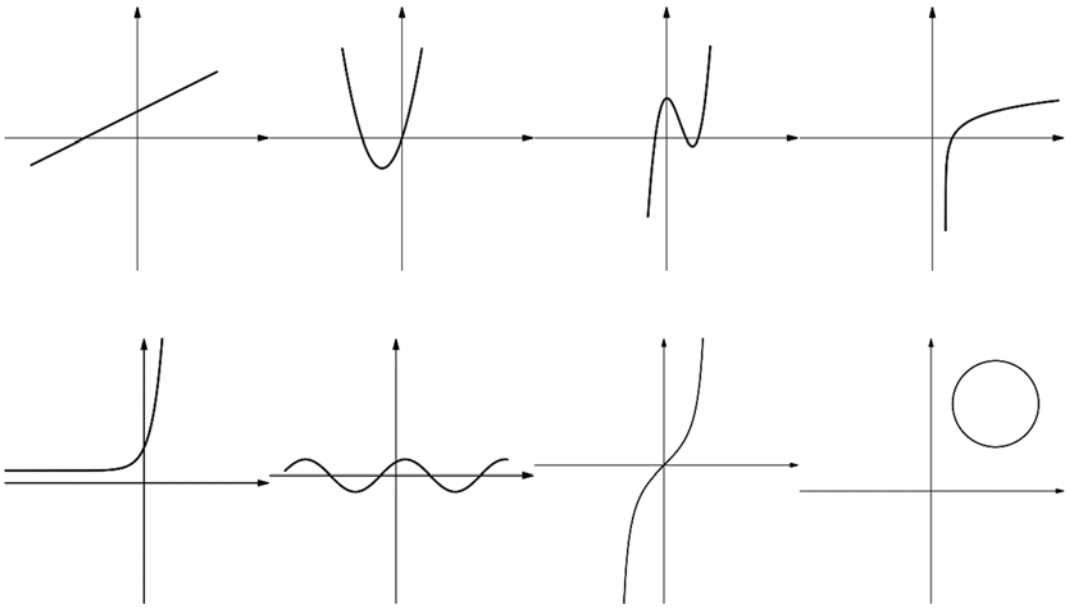


Figure 10. Part of the dataset is displayed.

The mathematical function graphs in the experiment come from the real book [42] function graphs and self-made high-quality function graphs, for a total of 1200. Some graph examples are shown in Figure 10. Each graph is composed of multiple graph elements, including horizontal axis, vertical axis, straight line, quadratic function curve, cubic function curve, sine function curve (contains sine, cosine, where the phase difference between sine and cosine is $\frac{\pi}{2}$), arcsine function curve (contains arcsine and arccosine, which are the inverse functions of sine and cosine), tan function curve, arctan function curve, exponential function curve, logarithmic function curve, composite function curve (free curve) and circle. For the labeling information such as equation and point coordinates in the function graph, the density method used in the paper of J. Chen et al. [26] is used to separate the information from the function entity, and only the function entity is retained.

In terms of processing, the graph is preprocessed first, including graph enhancement, flipping and size scaling. Second, Labelme software is used to annotate the graph elements in the graph. Polygon annotation method is used to annotate the contents, including the location and category information of graph elements. The corresponding graph elements annotation information is saved in JSON files. Then, the graph dataset is divided into the training set and the test set according to the ratio of 9:1.

The experimental environment of the computer was as follows: Intel Core i9-12900K 2.4–5.2 GHz CPU, 16 G RAM, Windows 10 64-bit operating system, Nvidia RTX 3090 Ti-24G, Python 3.7 and PyCharm 2022 as the programming platform.

Table 1. The main training parameters.

Define	Parameter	Annotation
IMAGE_MAX_DIM	1200	Picture size
IMAGE_MIN_DIM	400	Picture size
LEARNING_RATE	0.0001	Learning rate
NUM_CLASSES	13	Category
EPOCHS	150	Epochs
BATCH SIZE	8	Batch size
WEIGHT_DECAY	0.05	Weight decay

Table 1 shows the main training parameters used for the models in the experimental section of this study. To ensure the reliability and fairness of the experiments, this study maintains consistency in the main training parameters. In Table 1, the values for LEARNING_RATE and WEIGHT_DECAY are set based on empirical values. While LEARNING_RATE and WEIGHT_DECAY are empirical values, they have been proven effective through experimentation. In this experiment, a warm-up strategy was applied for the learning rate during the first 10 epochs, followed by the use of the Adam optimizer in subsequent epochs. These methods allowed for a smaller initial learning rate. Warm-up is a learning rate scheduling strategy that starts with a smaller learning rate in the initial training phase and gradually increases it to help the model converge better. The Adam optimizer combines momentum and adaptive learning rate adjustment methods, automatically adjusting the learning rate based on the gradients of the parameters.

WEIGHT_DECAY is used to prevent overfitting by adding a regularization term to the loss function. BATCH SIZE refers to the number of samples selected for each training iteration. In general, a larger BATCH SIZE is preferred as long as it does not lead to overgeneralization. However, in this experiment, the maximum BATCH SIZE is limited to 8 due to the available GPU memory. EPOCHS refers to the process of training the model using all the samples in the training dataset once. In this study, after conducting multiple experiments, it is observed that the total loss of each model starts to stabilize after reaching 150 epochs, showing no significant changes. Therefore, EPOCHS is set to 150.

4.2. Experimental results and analysis

The mathematical function graph data collected consists of the function graphs sampled from real books and the high-quality function graphs made by ourselves. In this section, the effectiveness of the proposed MA Mask R-CNN is demonstrated through experiments. The contents are presented in Tables 2 and 3. The categories of graph elements are as follows: C1 represents the horizontal axis, C2 represents the vertical axis, C3 represents a straight line, C4 represents a quadratic function, C5 represents a cubic function, C6 represents an exponential function, C7 represents a logarithmic function, C8 represents a sine function, C9 represents an arcsine function, C10 represents a tan function, C11 represents an arctan function, C12 represents a composite function, and C13 represents a circle. Tables 2 and 3 show the comparison between the model proposed in this study and other mainstream instance segmentation model algorithms. Among them, the evaluation index is mAP with IoU set at 0.75. Figures 11 and 12 show the visualization of results of Tables 2 and 3, respectively. Then, the mAP values of each model on mathematical function graphs target detection and segmentation are respectively shown in Tables 2 and 3, in which Mask R-CNN is the original model, YOLACT++ [43], PANET [44] and BlendMask [45] are classic models of instance segmentation, and Mask Dino [46] is the state-of-the-art instance segmentation model.

Table 2. mAP of each model on mathematical function graphs target detection.

Model	mAP (%)	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
Mask R-CNN	78.6	84.3	81.5	84.5	82.6	76.3	78.5	75.8	79.5	80.2	78.1	76.1	63.5	80.4
YOLACT++	72.0	76.2	74.2	71.6	70.3	77.6	72.1	69.4	65.8	74.2	71.8	70.1	64.1	79.1
PANET	80.3	86.7	84.9	86.5	83.1	78.2	79.4	77.1	81.1	81.9	79.4	78.3	65.5	81.4
BlendMask	82.2	87.1	87.5	86.9	87.1	82.3	80.6	77.5	81.3	83.9	80.1	81.2	69.4	84.1
Mask Dino	83.6	87.9	88.1	87.3	86.6	82.6	81.4	78.9	83.2	85.1	80.5	83.8	74.7	88.9
MA Mask R-CNN (ours)	**85.7**	**91.7**	**91.3**	**91.5**	**87.3**	**83.4**	**82.8**	**80.2**	**84.5**	**86.2**	**81.3**	**84.9**	**80.9**	**88.1**

The calculation method of the evaluation index mAP (mean Average Precision) in Table 2 is as follows:

Step 1: The region selection algorithm is used to get the candidate box. The IoU between each candidate box and the ground truth box is calculated. By comparing the IoU value and the given IoU threshold, the samples are classified as positive and negative samples, and a test set similar to that in the classification is obtained.

Step 2: The test set of Step 1 is used to calculate the score of positive samples through the classifier, and the positive and negative samples are classified by comparing the score value with the given score threshold.

Step 3: The accuracy rate and recall rate of the current category are calculated according to the results of Step 2.

Step 4: The AP value of each class is calculated, and the mAP is calculated according to the AP value of all the classes obtained.

As for the mAP in Table 3, one only needs to replace the IoU calculation target in Step 1 above with candidate mask and ground truth mask.

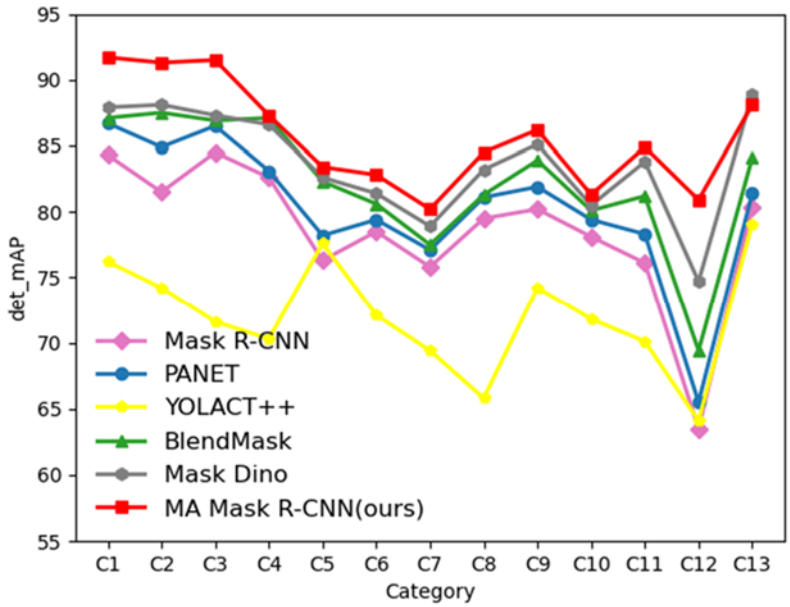


Figure 11. Object detection mAP of each method on each function class.

Combining Table 2 and Figure 11, it can be found that the target detection mAP of the model proposed in this study has been improved to varying degrees in each category. Especially, the target detection mAP of the C12 category (composite function) has been significantly improved. By introducing the attention module of local information and remote information, and strengthening the fusion of high-level and low-level information in the feature fusion stage, the MA Mask R-CNN model can effectively improve the detection ability of complex mathematical function graph recognition.

Furthermore, it can be observed that the model proposed in this study performs better than Mask Dino in all categories except for C13. There are several reasons for this: In the dataset used in this study, when annotating C13 (circle), a polygon annotation is used along the edge of the circle, and the entire solid circle is labeled as C13. Unlike other categories, C13 is not a thin line but a solid shape. Mask Dino has a significant advantage in handling non-elongated objects like C13. Therefore, the model proposed in this study is relatively weaker in the C13 category compared to Mask Dino. It is important to consider the specific characteristics and challenges posed by different object categories when evaluating the performance of models. While our model may not outperform Mask Dino in the C13

category due to the nature of the annotations, it demonstrates superior performance in other categories.

Table 3. mAP of each model on mathematical function graphs segmentation.

Model	mAP(%)	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
Mask R-CNN	64.1	73.1	71.6	65.3	73.6	70.5	56.5	54.3	30.5	66.7	56.2	60.8	76.2	77.5
YOLACT++	59.4	67.1	66.8	68.2	65.3	65.1	57.4	55.1	32.6	60.6	51.6	58.1	50.8	73.9
PANET	66.5	75.3	72.1	69.6	75.6	71.1	58.1	56.2	35.8	69.8	58.4	63.1	78.1	80.8
BlendMask	69.4	75.8	71.9	74.6	76.1	74.2	61.3	62.1	46.7	73.8	61.5	65.2	77.8	81.1
Mask Dino	71.2	76.1	73.2	76.9	76.1	74.7	61.8	63.1	58.6	74.9	61.9	67.3	78.1	83.1
MA Mask R-CNN (ours)	**72.3**	**77.4**	**75.4**	**78.1**	**76.9**	**75.8**	**62.1**	**63.4**	**62.7**	**75.8**	**63.3**	**68.5**	**78.1**	**82.7**

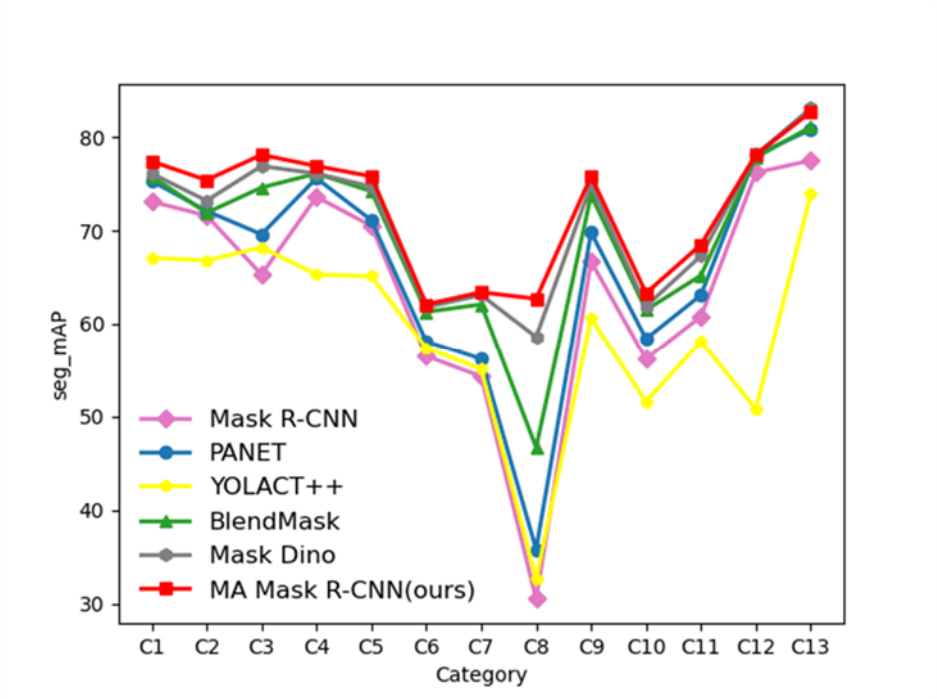


Figure 12. mAP of each category in segmentation branch.

Combined with Table 3 and Figure 12, it can be found that the model proposed in this study has different enhancement degrees of segmentation mAP in each category. Among them, the improvement of our proposed model compared to the state-of-the-art instance segmentation model, Mask Dino, is limited in categories such as exponential functions C6, logarithmic functions C7, and composite functions C12. Additionally, the improvement in category C13 is slightly inferior to that of Mask Dino. Potential reasons are as follows: Mask Dino has an advantage in detecting the C13 category. Furthermore, its utilization of query embeddings for performing dot products on high-resolution pixel embedding maps enables the prediction of a set of binary masks. This approach proves particularly effective for the C13 category, which exhibits fixed geometric shapes without elongation. It allows for a better understanding of the shape information of the target and generates

masks that are more compatible with the shape of C13. Therefore, Mask Dino still maintains a certain advantage in functionally segmenting images of the C13 category. In some cases, C6 and C7 have high morphological similarity, and the feature difference is not obvious. However, due to the particularity of the category, C12 has no fixed morphological rules, and some entities easily have similarities with other categories, relying more on the supplement of datasets, so it may be due to insufficient training samples and the graph elements of some training samples extending to the edge of the image, which affects the segmentation. Due to the wide range of independent variables of sine function, more features, and higher requirement for continuity than other categories in samples, the remote information acquisition capability of models is also higher, resulting in limited improvement of all preceding models on C8 (sine function). However, our model has obvious improvement in this category, and the reasons are as follows: The improvement of branch detection effect, the expansion of the anchor frame ratio in the RPN layer and the addition of penalty factor to the IoU make the candidate frame more accurate and bring some elastic space. Benefiting from the foundation of the above gain, the PointRend of segmentation based on graph rendering ideas can better improve the segmentation accuracy and generate high-quality segmentation masks.

The backbone network of Mask R-CNN, YOLACT++, PANET, BlendMask and Mask Dino in Tables 2 and 3 is ResNet101. By comparison, the MA Mask R-CNN model proposed in this study is significantly improved in both target detection mAP and segmentation mAP. Compared with the original Mask R-CNN, there are increases by 9% in the mAP of target detection and by 12.8% in the mAP of segmentation. Moreover, it also has certain advantages over Mask Dino. MA Mask R-CNN model improved on the mAP but failed to achieve the desired result. The reasons are as follows: 1) The convolution of a common fixed grid position can only extract the features of a rectangular box, which lacks geometric transformation ability and cannot better obtain the features of function graphs. 2) Insufficient dataset samples resulted in insufficient features learned by the model. 3) The noise of the dataset is too small, which makes the anti-interference ability of the model insufficient.

Table 4. Ablation results.

Model	Params	Det_mAP (%)	Seg_mAP (%)	GFLOPs	FPS (f/s)
Mask R-CNN	43.9 M	78.6	64.1	251.4	5.4
+MA ConvNeXt	51.3 M	81.8(+3.2)	66.8(+2.7)	271.1	4.9
+MA BiFPN	60.8 M	85.1(+3.3)	68.6(+1.8)	296.2	4.6
+RPN improvement	61.2 M	85.7(+0.6)	69.7(+1.1)	296.2	4.6
+PointRend	61.2 M	85.7(+0.0)	72.3(+2.6)	296.2	4.6

Table 4 shows the ablation experimental results of this study, in which Params denotes the number of parameters contained in the model (unit: MB), abbreviated as M in the table. Det mAP denotes the mAP for object detection, and Seg mAP denotes the mAP for segmentation. GFLOPs are billion floating-point operations per second and are used to measure the complexity of a model. FPS denotes the number of graphs the model can process per second. Table 5 displays the runtime of each model mentioned in the experimental section of this study, in which Time denotes the duration from when the input data undergoes preprocessing and is fed into the model to start the timer, until the

model generates the output results and the timer stops. According to the analysis of Table 4, compared with the original Mask R-CNN, the model proposed in this study has significantly improved the mAP of target detection and the mAP of instance segmentation. According to the analysis of Tables 4 and 5, compared with the original Mask R-CNN and Mask Dino, the accuracy of the model proposed in this study is improved at the cost of less speed loss. Therefore, the ablation results prove the effectiveness of the improved model in this study.

Table 5. Runtime of each model

	Mask R-CNN	YOLACT++	PANET	BlendMask	Mask Dino	MA Mask R-CNN (ours)
Time (s)	2.16	1.27	2.29	2.37	2.49	2.63

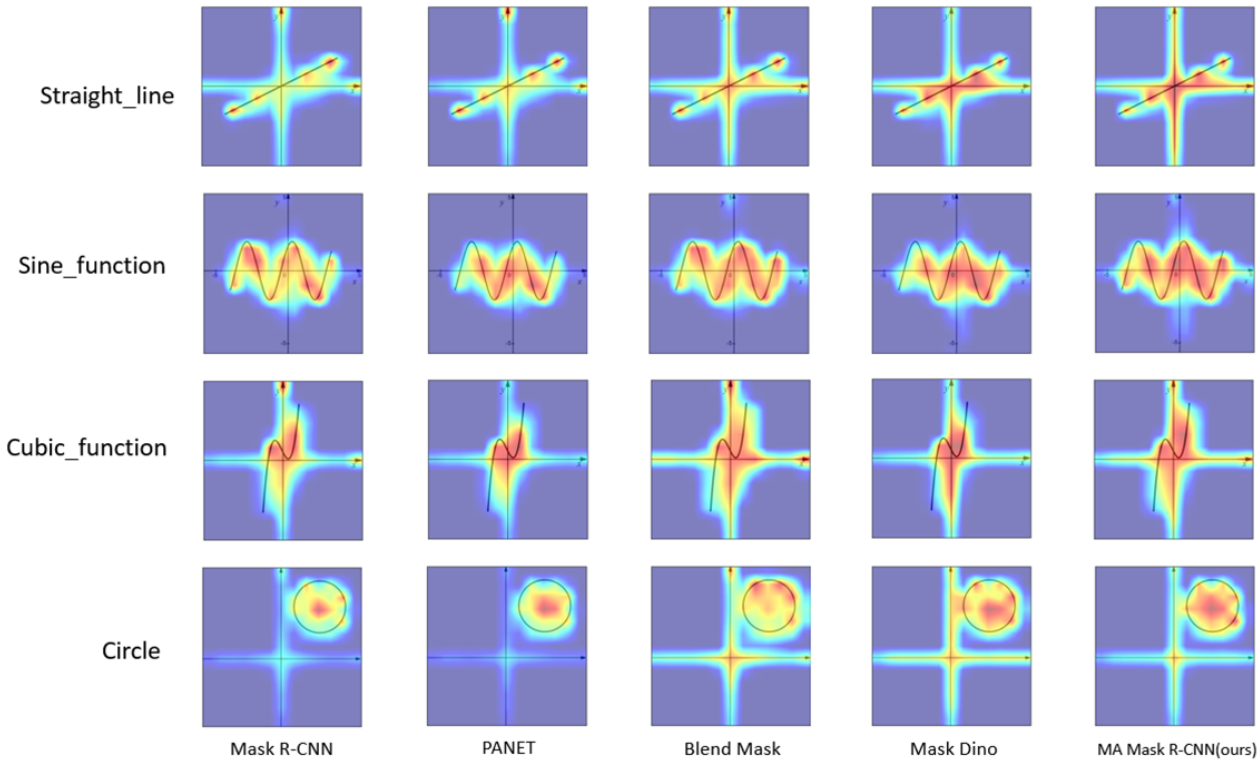
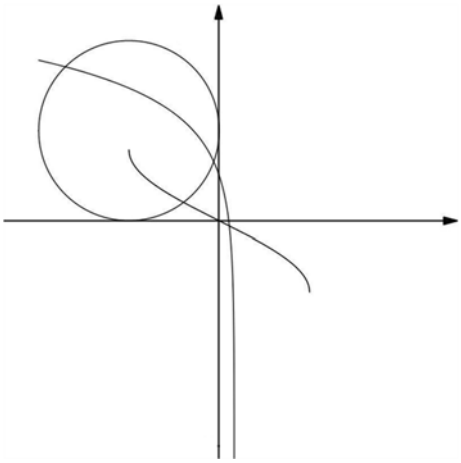
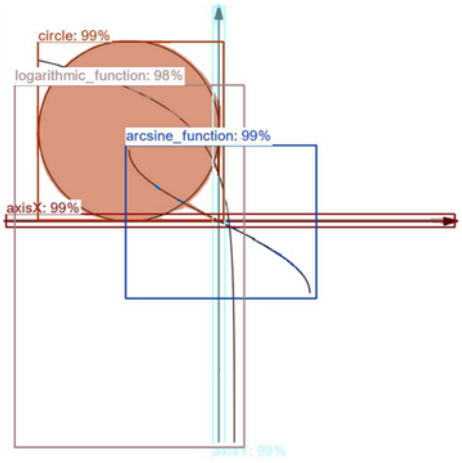


Figure 13. Different CAM visualization results of Grad-CAM [47] on Mask R-CNN, PANET, Blend Mask, MA Mask R-CNN (ours) are used to compare the attention of different models to graphs.

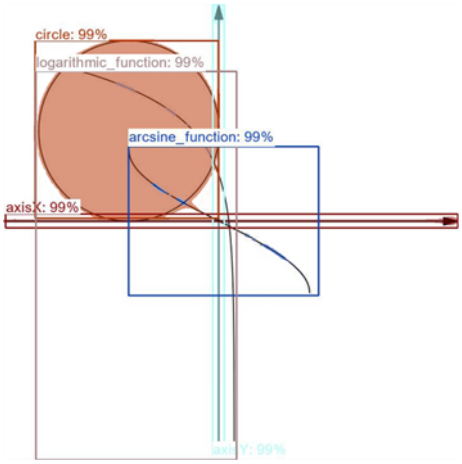
This section does not compare YOLACT++ because it only has a clear advantage in speed. CAM (Class Activation Mapping) is a common visualization tool for attention mapping. By analyzing Figure 13, it can be found that compared with Mask R-CNN, PANET and Blend Mask models, MA Mask R-CNN can pay more attention to target graphs in various mathematical function graphs. In addition, compared with Mask Dino, MA Mask R-CNN only pays less attention to circle. So, MA Mask R-CNN has better performance in mathematical function graphs, which proves that this model is effective in improving local attention and remote information.



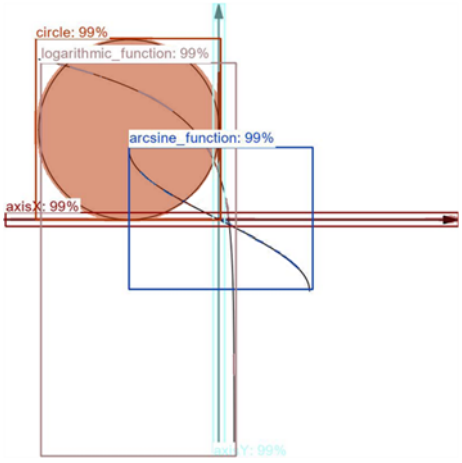
(a) Original graphs (composed of coordinate axes, logarithmic function, arcsine function and circle)



(b) Mask R-CNN

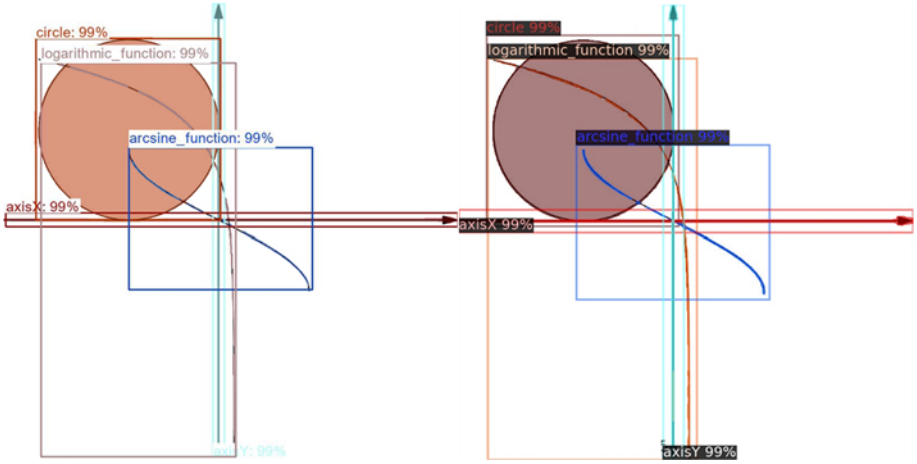


(c) PANET



(d) Blend Mask

Continued on next page

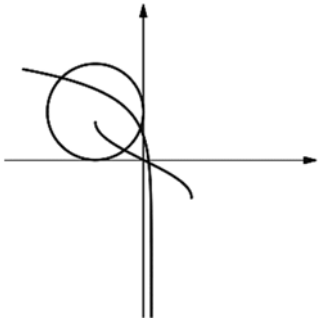


(e) Mask Dino (e) MA Mask R-CNN(Ours)

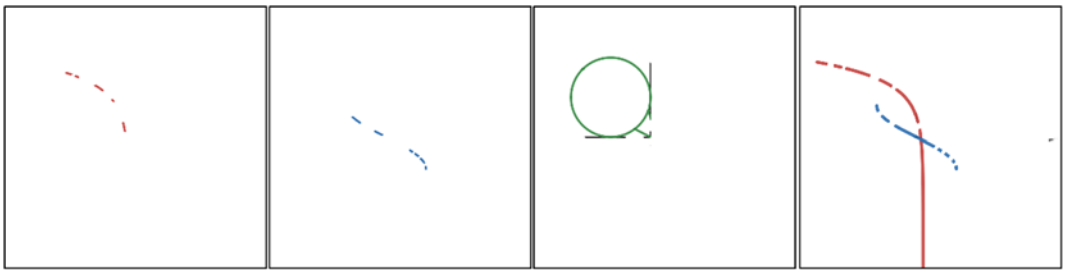
Figure 14. Instance segmentation effect of Mask R-CNN, PANET, Blend Mask, Mask Dino, MA Mask R-CNN (ours).

Figure 14 shows the instance segmentation results of the four models on the test graph. In order to better display the recognition and segmentation effect of each model on the function graph in Figure 14, the segmentation instances of each model in the original Figure 14 are extracted separately in Figure 15. In Figure 15(b)–(f), the red line denotes the solid part of the logarithmic function, the blue line denotes the solid part of the arcsine function, the green line denotes the solid part of the circle, and the black line denotes the solid part of the coordinate axis. Among them, each first image in Figure 15(b)–(f) denotes the logarithmic function divided by the model from Figure 15(a), the second image in Figure 15(b)–(f) denotes the arcsine function entity divided by Figure 15(a), the third image in Figure 15(b)–(f) denotes the circle divided by the Figure 15(a), and the fourth image in Figure 15(b)–(f) denotes the remaining part of the model after removing the entity divided by the first three graphs from Figure 15(a). The following is an analysis of the instance segmentation results of the four models refined in Figure 15 (the four models are similar in the segmentation effect of the axes, so the analysis is not carried out): 1) The logarithmic function and arcsine function entities extracted by the original Mask R-CNN segmentation in Figure 15(b) are too few, with poor continuity. Although a complete circle is divided through the mask of the circle, part of the coordinate axis and the entity of the arcsine function are also incorrectly divided. Through the analysis of the fourth image in Figure 15(b), it can be found that the instances that Mask R-CNN can extract are limited, the features of the residual instances are not fully utilized, and the segmentation effect of the junctions of different instances is poor. 2) The segmentation effect of PANET in Figure 15(c) is slightly improved. By analyzing the fourth image in Figure 15(c), it can be found that there is still a large amount of entity information that has not been utilized. 3) In Figure 15(d), the logarithmic function and arcsine function entities extracted by Blend Mask have been significantly improved. The interference of the arcsine function instance can be removed in the circle segmentation, but the partial axis entity intersecting with the circle boundary is still incorrectly segmented. By analyzing the fourth image in Figure 15(d), it can be found that the Blend Mask has obtained more entity information than the previous two models, but the utilization rate of entity information still cannot meet the requirements. 4) In Figure 15(e), it can be observed that Mask Dino exhibits further improvement in extracting various function entities compared to Blend Mask. The function entities

become more complete, but there are still instances where the logarithmic function and arcsine function suffer from mutual occlusion with other functions, resulting in the inability to generate masks for the occluded portions. However, Mask Dino shows the most significant improvement in segmenting circle function entities by eliminating the interference caused by neighboring function entities and coordinate axes, resulting in a segmented mask that only contains the circular function. Analyzing the fourth image in Figure 15(e), it can be noticed that Mask Dino has a higher utilization rate of function entity information compared to the previous three models. However, there is still room for improvement. 5) In Figure 15(f), the logarithmic function and arcsine function entities segmented by MA Mask R-CNN (ours) appear to be the most complete and continuous. Additionally, its performance on circle is nearly comparable to Mask Dino, as it effectively eliminates the interference from other function entities when segmenting the circle. Moreover, it successfully avoids segmenting the coordinate axis entities at the intersection with the circle as part of the circle's mask. By analyzing the fourth image in Figure 15(f), it can be found that MA Mask R-CNN (ours) has the least remaining entities after removing each segmentation instance, indicating that this model makes the best use of the instance information of each function graph. From the above, it can be found that the model proposed in this study can significantly improve the function instance segmentation effect by strengthening the multi-scale fusion of local information and global information and adopting the mask segmentation method, which can generate higher-quality and smoother results. This model can generate more complete and more continuous masks on the function graphs, and the masks can be generated correctly at the classifications of different function categories.

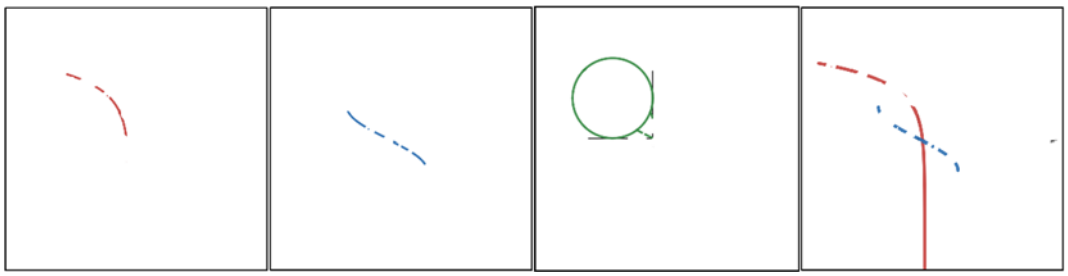


(a) Original graph (composed of coordinate axes, logarithmic function, arcsine function and circle).

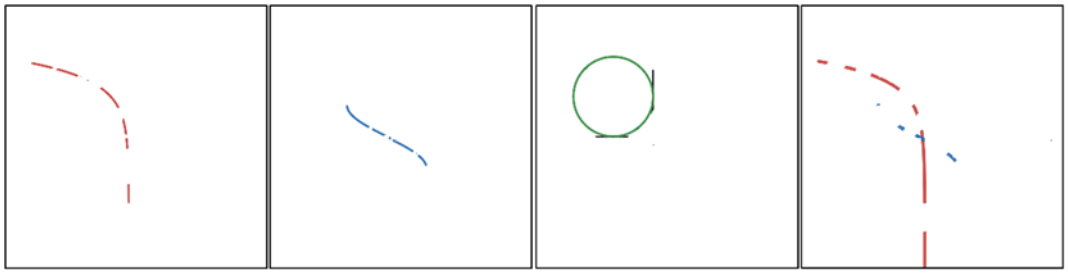


(b) Mask R-CNN segments the function instance of the original graph according to the mask of each function generated.

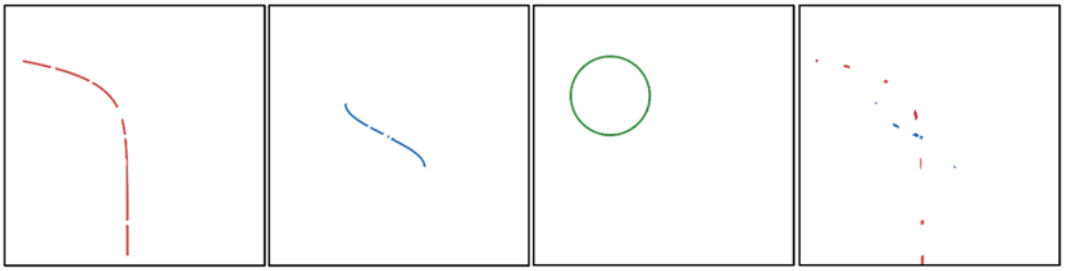
Continued on next page



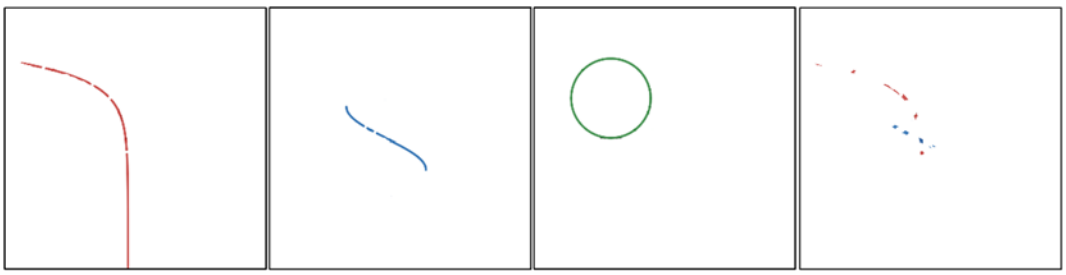
(c) PANET segments the function instance of the original graph according to the mask of each function generated.



(d) Blend Mask segments the function instance of the original graph according to the mask of each function generated.



(e) Mask Dino segments the function instance of the original graph according to the mask of each function generated.



(f) MA Mask R-CNN (ours) segments the function instance of the original graph according to the mask of each function generated.

Figure 15. Breakdown diagram of instance segmentation effect of Mask R-CNN, PANET, Blend Mask, MA Mask R-CNN (ours) on Figure 15(a).

5. Conclusions and prospect

In this paper, MA Mask R-CNN is proposed to improve the detection and segmentation capabilities of the original Mask R-CNN model for mathematical function graphs. Improvements include the following: 1) adding an attention module with a combination of local and remote information, using ConvNeXt as a baseline; 2) taking BiFPN as the baseline, adding RFA and ASPP modules to strengthen the fusion of multi-scale information at high and low layers of BiFPN; 3) adding penalty factor to IoU in RPN stage, allowing candidate anchor box to have a certain redundancy, so that the detection box can better contain categories and enhance the segmentation effect; 4) adopting PointRend mask segmentation strategy, according to the rendering angle of computer graphs, making the segmentation results smoother and more complete. Moreover, experimental results show that MA Mask R-CNN is better than other models in case segmentation of function graphs.

In the following directions, we will continue to improve the work of mathematical function graph instance segmentation technology in the production of tactile graphics:

1) Improve the detection speed of the model and continue to reduce the parameters. Pruning the model allows the model to be lightweight and quickly applied to different scenarios.

2) Expand the dataset and optimize the distribution of the dataset. The classification of the dataset can still be optimized. The generalization ability of the model is gradually strengthened by expanding the well-distributed dataset.

3) Adding Deformable convolution [48] to enhance the geometric transformation ability of the model, so that the model can extract more function image features.

4) After segmenting the function graph, the mask is skeletonized, and the resulting subdivision mask is redrawn through fitting. Finally, the fitted mathematical function graph is saved in SVG format, providing a foundation for the subsequent production of tactile graphics.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62076111, 62006099), the Key Research and Development Program of Zhenjiang-Social Development (Grant No. SH2018005), the Natural Science Foundation of Jiangsu Higher Education (Grant No. 17KJB520007), Industry-school Cooperative Education Program of the Ministry of Education (Grant No. 202101363034).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. World Health Organization, *World Report on Vision*, 2019. Available from: https://www.who.int/publications/i/item/world-report-on-vision.
2. K. Han, Y. Liu, F. Yuan, The tactile graphics representation and its design application for the visually impaired, *Design*, **4** (2015), 18–19.

3. D. Wu, J. Gan, Assistant haptic interaction technology for blind Internet user, *J. Eng. Design*, **17** (2010), 128–133.
4. X. Li, Reading resources for the blind and the present situation of reading promotion in China, *New Century Lib.*, **5** (2013), 19–22.
5. S. Spooner, "What page, Miss?" enhancing text accessibility with DAISY (Digital Accessible Information System), *J. Visual Impairment Blindness*, **108** (2014), 201–211. https://doi.org/10.1177/0145482X1410800304
6. S. Suksakulchai, S. Chaisanit, A daisy book production system over the internet, *J. Commun. Comput.*, **8** (2011), 744–750. https://doi.org/10.1038/sj.bdj.4800904
7. R. Zhang, Digital accessible information system and its development trend, *Discovering Value*, **6** (2010), 304.
8. A. Quint, Scalable vector graphics, *IEEE Multimedia*, **10** (2003), 99–102. https://doi.org/10.1109/MMUL.2003.1218261
9. G. Browne, Ebook indexes, EPUB and the international digital publishing forum, *Online Curr.*, **26** (2012), 127–130.
10. P. P. Rege, A. C. Chandrakar, Text-image separation in document images using boundary/perimeter detection, *ACEEE Int. J. Signal Image Process.*, **3** (2012), 10–14. https://doi.org/01.IJSIP.03.01.70
11. P. Lyu, M. Liao, C. Yao, W. Wu, X. Bai, Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), 67–83. https://doi.org/10.48550/arXiv.1807.02242
12. Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2019), 9365–9374. https://doi.org/10.1109/CVPR.2019.00959
13. J. Memon, M. Sami, R. A. Khan, M. Uddin, Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR), *IEEE Access*, **8** (2020), 142642–142668. https://doi.org/10.1109/ACCESS.2020.3012542
14. T. Hegghammer, OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment, *J. Comput. Social Sci.*, **5** (2022), 861–882. https://doi.org/10.1007/s42001-021-00149-1
15. X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, et al., East: an efficient and accurate scene text detector, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 5551–5560. https://doi.org/10.1109/CVPR.2017.283
16. M. Suzuki, T. Kanahori, N. Ohtake, K. Yamaguchi, An integrated OCR software for mathematical documents and its output with accessibility, in *International Conference on Computers for Handicapped Persons*, (2004), 648–655. https://doi.org/10.1007/978-3-540-27817-7_97
17. M. Suzuki, *Pen-Based Mathematical Computing Organizers*, Stephen Watt and Clare So, (2005), 60.
18. Z. Wang, J. C. Liu, Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training, *Int. J. Doc. Anal. Recogn.*, **24** (2021), 63–75. https://doi.org/10.1007/s10032-020-00360-2
19. O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2015), 3156–3164. https://doi.org/10.1109/CVPR.2015.7298935

20. Y. Wang, J. Xu, Y. Sun, End-to-end transformer based model for image captioning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **36** (2022), 2585–2594. https://doi.org/10.1609/aaai.v36i3.20160
21. T. Fuda, S. Omachi, H. Aso, Recognition of line graph images in documents by tracing connected components, *Syst. Comput. Jpn.*, **38** (2007), 103–114. https://doi.org/10.1002/scj.10615
22. S. Shimada, S. Kakumoto, M. Ejiri, A recognition algorithm of dashed and chained lines for automatic inputting of drawings, *IEICE Trans. Inf. Syst.*, **18** (1986), 759–770. https://doi.org/10.1002/scj.4690180603
23. N. Yokokura, T. Watanabe, Recognition of business bar-graphs using layout structure knowledge, *J. Inf. Process.*, **40** (1999), 2954–2966.
24. M. Yo, T. Kawahara, R. Fukuda, Handwriting graphics input system for high school mathematics, *IEICE Tech. Rep.*, **291** (2004), 43–48.
25. A. Balaji, T. Ramanathan, V. Sonathi, Chart-text: A fully automated chart image descriptor, preprint, arXiv:1812.10636. https://doi.org/10.48550/arXiv.1812.10636
26. J. Chen, N. Takagi, Development of a method for extracting and recognizing graph elements in mathematical graphs for automating translation of tactile graphics, *J. Jpn. Soc. Fuzzy Theory Intell. Inf.*, **26** (2014), 593–605. https://doi.org/10.3156/jsoft.26.593
27. J. Staker, K. Marshall, R. Abel, C. M. McQuaw, Molecular structure extraction from documents using deep learning, *J. Chem. Inf. Model.*, **59** (2019), 1017–1029. https://doi.org/10.1021/acs.jcim.8b00669
28. M. Oldenhof, A. Arany, Y. Moreau, J. Simm, ChemGrapher: optical graph recognition of chemical compounds by deep learning, *J. Chem. Inf. Model.*, **60** (2020), 4506–4517. https://doi.org/10.1021/acs.jcim.0c00459
29. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, in *Advances in Neural Information Processing Systems*, **30** (2017), 5998–6008. https://doi.org/10.48550/arXiv.1706.03762
30. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al., An image is worth 16x16 words: Transformers for image recognition at scale, preprint, arXiv:2010.11929. https://doi.org/10.48550/arXiv.2010.11929
31. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, et al., Swin transformer: Hierarchical vision transformer using shifted windows, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2021), 10012–10022. https://doi.org/10.1109/ICCV48922.2021.00986
32. Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2022), 11976–11986. https://doi.org/10.1109/CVPR52688.2022.01167
33. M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01079
34. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision*, **42** (2017), 2961–2969. https://doi.org/10.1109/TPAMI.2018.2844175
35. L. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, **40** (2018), 801–818. https://doi.org/10.1109/TPAMI.2017.2699184

36. J. Liu, W. Zhang, Y. Tang, J. Tang, G. Wu, Residual feature aggregation network for image super-resolution, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 2359–2368. https://doi.org/10.1109/CVPR42600.2020.00243
37. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in *Advances in Neural Information Processing Systems*, (2015), 28. https://doi.org/10.48550/arXiv.1506.01497
38. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
39. T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2017), 936–944. https://doi.org/10.1109/CVPR.2017.106
40. A. Kirillov, Y. Wu, K. He, R. Girshick, PointRend: Image segmentation as rendering, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 9799–9808. https://doi.org/10.1109/CVPR42600.2020.00982
41. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 2818–2826. https://doi.org/10.1109/CVPR.2016.308
42. Cambridge International Examinations, IGCSE Guide Additional Mathematics Examples & Practice Book 1, *Curriculum Planning and Development*, 2021.
43. D. Bolya, C. Zhou, F. Xiao, Y. J. Lee, Yolact++: Better real-time instance segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2020). https://doi.org/10.1109/TPAMI.2020.3014297
44. S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 8759–8768. https://doi.org/10.1109/CVPR.2018.00913
45. H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, Y. Yan, Blendmask: Top-down meets bottom-up for instance segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 8573–8581. https://doi.org/10.1109/CVPR42600.2020.00860
46. F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, et al., Mask dino: Towards a unified transformer-based framework for object detection and segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2022), 3041–3050. https://doi.org/10.48550/arXiv.2206.02777
47. R. R. Selvaraju, M. Cogswell, A. Das, R. V edantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in *Proceedings of the IEEE International Conference on Computer Vision*, (2017), 618–626. https://doi.org/10.1007/s11263-019-01228-7
48. X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2019), 9308–9316. https://doi.org/10.1109/CVPR.2019.00953

AIMS Press