IEMask R-CNN: Information-Enhanced Mask R-CNN

Xiuli Bi, Jinwu Hu, *Student Member, IEEE*, Bin Xiao, Weisheng Li, *Member, IEEE*, and Xinbo Gao, *Senior Member, IEEE*

Abstract—The instance segmentation task is relatively difficult in computer vision, which requires not only high-quality masks but also high-accuracy instance category classification. Mask R-CNN has been proven to be a feasible method. However, due to the Feature Pyramid Network (FPN) structure lack useful channel information, global information and low-level texture information, and mask branch cannot obtain useful local-global information, Mask R-CNN is prevented from obtaining high-quality masks and high-accuracy instance category classification. Therefore, we proposed the Information-enhanced Mask R-CNN, called IEMask R-CNN. In the FPN structure of IEMask R-CNN, the information-enhanced FPN will enhance the useful channel information and the global information of the feature maps to solve the issues that the high-level feature map loses useful channel information and inaccurate of instance category classification, meanwhile the bottom-up path enhancement with adaptive feature fusion will ultilize the precise positioning signal in the lower layer to enhance the feature pyramid. In the mask branch of IEMask R-CNN, an encoding-decoding mask head will strength local-global information to gain a high-quality mask. Without bells and whistles, IEMask R-CNN gains significant gains of about 2.60%, 4.00%, 3.17% over Mask R-CNN on MS COCO2017, Cityscapes and LVIS1.0 benchmarks respectively.

Index Terms—Instance segmentation, information-enhanced FPN, adaptive feature fusion, encoding-decoding mask head

1 Introduction

Deep Convolutional Neural Networks are dramatically driving the development of computer vision [1], [2], leading to a series of latest tasks including classification [3], [4], objection detection [5], [6], [7], semantic segementation [8], [9], [10], etc. Classification is to distinguish different types of images based on the semantic information of the image. It is a basic and important problem in computer vision, and it is also the basis for other high-level vision tasks such as object detection and semantic segmentation. Object detection aims to accurately predict the semantic category and the location described by a bounding box for each object instance, which is quite a coarse localization. Unlike object detection, the semantic segmentation task aims to assign the pixel-wise labels for each image while providing no indication of the object instances, such as the number of object instances or the specific semantic region for any particular instance. Compared with classic class-level semantic segmentation or bounding box-level object detection, instance segmentation provides in-depth understanding by distinguishing different object instances at the pixel level, widely benefiting autonomous vehicles, robotics, video surveillance, etc. Therefore, instance segmentation has become one of the important, complex and challenging fields in machine vision research.

The instance segmentation task is a relatively difficult one among computer vision tasks. It requires not only high-quality masks, but also high-accuracy instance category classification. Current state-of-the-art (SOTA) solutions to this challenging task can be classified into proposal-based [11], [12], [13] and proposal-free methods [14], [15], [16]. Some important content of different SOTA instance segmentation methods are shown in Table 1. The proposal-free instance segmentation methods predict the category labels of each pixel first and then group them together to form instance segmentation results. Although the proposal-free instance segmentation method can predict high-quality masks, the accuracy of the instance segmentation task is generally low because of its poor instance discrimination ability. In addition, most proposal-free instance segmentation methods use cumbersome post-processing methods [17], [18], and their generalization ability is too poor to cope with complex scenes with many categories.

Compared with proposal-free instance segmentation method, proposal-based method exploit the state-of-the-art detectors, such as Faster R-CNN [6], which gets the region of each instance and then predicts the mask for each region. Therefore, although the proposal-based method cannot obtain the high-quality masks, since the proposal-based method is extended based on the objection detection task, it can distinguish different instances well and obtain better instance segmentation results. In addition, the proposal-based method has strong generalization ability and can handle multiple types of complex scenes. The proposal-based method has more advantages thus attract more attention, achieving a rapid

• *The authors are with the Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. E-mail: {bixl, xiaobin, liws, gaoxb}@cqupt.edu.cn, s200231140@stu.cqupt.edu.cn.*

Manuscript received 22 March 2022; revised 24 May 2022; accepted 27 June 2022. Date of publication 30 June 2022; date of current version 14 March 2023.
This work was supported in part by National Key Research and Development Project under Grants 2019YFE0110800 and 2016YFC1000307-3, in part by the National Natural Science Foundation of China under Grants 62172067 and 61976031, in part by the National Major Scientific Research Instrument Development Project of China under Grant 62027827.
(Corresponding author: Bin Xiao.)
Recommended for acceptance by C. Yang.
Digital Object Identifier no. 10.1109/TBDATA.2022.3187413

TABLE 1
Some Important Content of Different SOTA Instance Segmentation Methods

	Method	Important Content
Proposal-free	Liang *et al.* [16]	Use spectral clustering to cluster the pixels
	Kirillov *et al.* [19]	Add boundary detection information during the clustering procedure
	Bai *et al.* [15]	Predict pixel-level energy values and use watershed algorithms for grouping
	Newell *et al.* [20]	Use metric learning to learn the embedding
	Pei *et al.* [14]	Develop an adaptive quantile strategy to flexibly and automatically select the initial clustering centers
Proposal-based	Bolya *et al.* [11]	Add the Mask branch to YOLO
	Tian *et al.* [21]	Processing instance using dynamic conditional convolution
	He *et al.* [22]	Add Mask head on Faster R-CNN
	Huang *et al.* [13]	Add a branch for mask IoU score prediction based on the mask prediction on the original Mask R-CNN
	He *et al.* [12]	Adaptive selection region by an iterative algorithm, the pixels within the region and to fine-tune the predicted
	Huang *et al.* [23]	Incorporate boundary information to guide the mask learning for better boundaries and localization
	Shen *et al.* [24]	Use a mask represented by DCT to achieve a high-resolution and low-complexity prediction mask

development. The Mask R-CNN shown in Fig. 1 is the most classic model in the proposal-based method, which uses the FCN branch to extend Faster R-CNN to segment objects in the detection frame. Mask R-CNN has been proven to be an effective method for instance segmentation, but there are still some issues in the structure of Mask R-CNN that are worth exploring. The issues of Mask R-CNN can be summarized as follows.

First, in the top-down propagation process of FPN, stronger semantic information obtained from high-level features can be used to improve low-level features, but high-level features lose too much useful channel information, such as directly using 1×1 convolution to integrate the backbone with the features of 2048 channels are compressed into features of 256 channels [25]. And the feature maps generated by the FPN structure are limited by the lack of global information, which in turn affects the performance of downstream classification tasks. Second, the information of low-level feature maps can help improve the localization ability of high-level feature maps. However, the FPN structure has a long path from low-level structure to topmost features, which increases the difficulty of high-level feature maps of FPN to access the accurate localization information [26]. Third, in the mask branch of Mask R-CNN, there are only four consecutive 3×3 convolution operations on ROI aligned feature maps, which misses the integration of local and global information and result in an inaccurate final mask [22], [27].

In response to the above-mentioned issues of Mask R-CNN, we focused on the classic model Mask R-CNN and proposed the Information-enhanced Mask R-CNN, called IEMask R-CNN. Compared with the existing Mask R-CNN series models, the main contributions of IEMask R-CNN are generalized as follows:

- Information-enhanced FPN we designed, which effectively enhances the useful channel information of high-level feature maps and effectively improves the performance of downstream classification tasks.
- The bottom-up path enhancement with adaptive feature fusion in FPN we designed, which effectively enhances the positioning capabilities of high-level feature maps.
- For the first time, we proposed encoding-decoding mask head to obtain local-global information, and then to obtain the high-quality masks.
- We conducted extensive experiments to show that our IEMask R-CNN can not only produce high-quality masks, but also improve the more accuracy of instance classification.

The rest of this paper is organized as follows. Related work is introduced in Section 2. The proposed IEMask R-CNN is introduced in detail in Section 3. Section 4 provides the experiments and discussions. Finally, we concluded this paper in Section 5.

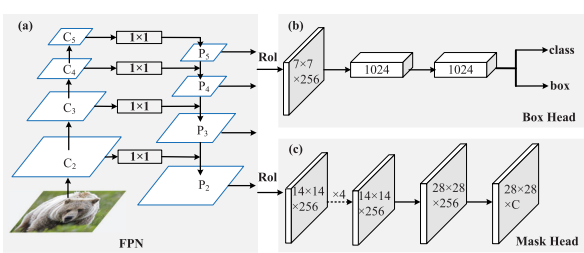

Fig. 1. The overall architecture of Mask R-CNN (RPN has been omitted.).

2 Related Work

Instance segmentation is relatively a difficult task in computer vision, which requires both the same pixel-level classification as semantic segmentation and part of the characteristics of real target detection, i.e., the need to locate different instances, even though they belong to the same class. After that, there are two main lines in study of instance segmentation: proposal-free and proposal-based instance segmentation.

2.1 Proposal-Free Instance Segmentation

The proposal-free instance segmentation methods are mainly based on the morphology and spatial relationship of all the objects in the images, and this type of method first predicts the category label of each pixel and then groups them to form

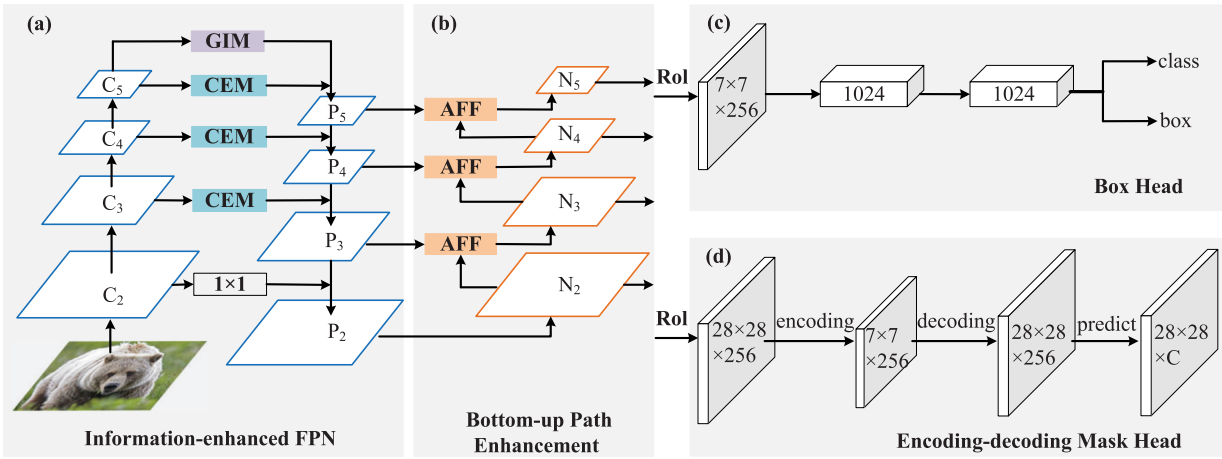

Fig. 2. Illustration of our IEMask R-CNN. (a) Information-enhanced FPN. (b) Bottom-up path Enhancement. (c) Box Head. (d) Encoding-decoding Mask Head. (RPN has been omitted.).

the instance segmentation result. For example, object boundary is an important feature for separating the touching object. Bai *et al.* [15] distinguish instances based on probability maps of foreground objects and their boundaries. [28] separates each instance according to the distance between the two connected components. Additionally, post-processing methods are employed to separate the touching objects based on the semantic segmentation predictions, such as conditional region growing algorithm [29] and watershed algorithm [17], [18]. Zhang *et al.* [18] proposed an image-to-image translation method for a more accurate probability map compared with the classification-based method.

2.2 Proposal-Based Instance Segmentation

Compared with the proposal-free methods, most of the proposal-based methods have better generalization ability and accuracy. Most of the proposal-based methods generate masks after locating objects with bounding boxes generated by detectors [6], [30], [31], and Mask R-CNN [22] is one of the fundamental proposal-based instance segmentation methods. Based on the high-dimensional feature maps from the backbone CNN network, Mask R-CNN first generates regions of interest (ROIs) containing the foreground objects with a region proposal network (RPN). After aligning the ROIs to the same size, a box sub-branch and a mask sub-branch are employed to predict the coordinate, class label, and mask prediction for each ROI. With the help of the local level information from the spatial locations of the instances, Mask R-CNN achieved SOTA performance compared with the traditional box-free methods.

Following the Mask R-CNN, many methods were further proposed with higher accuracy. To achieve high performance and multi-scale feature representation, Feature Pyramid Network (FPN) [32] is utilized to extract in-network feature hierarchy, where a top-down path with lateral connections is augmented to propagate semantically strong features. At present, the Mask R-CNN structure with FPN has become a mainstream method. FPN structure [32] pools feature from all feature levels and shorten the distance among lower and topmost feature levels for reliable information passing. Huang *et al.* [13] added a branch for mask IoU score prediction based on the mask prediction on the original Mask R-CNN. Cai *et al.* [33] employed a cascade connection of several bounding boxes and mask prediction sub-branches. Cheng *et al.* [23] incorporates boundary information to guide the mask learning for better boundaries and localization. Kirillow *et al.* [12] through an iterative algorithm to adaptively select the problematic area, and fine-tune and predict the pixels in the area. Shen *et al.* [24] proposed to use a mask represented by DCT to achieve a high-resolution and low-complexity prediction mask.

Different from the previous Mask R-CNN series methods, our IEMask R-CNN takes the FPN structure and mask branch of Mask R-CNN as the research point, and proposes information-enhanced FPN, bottom-up path enhancement and encoding-decoding mask head. Our IEMask R-CNN can take advantage of useful information channels and local-global information, not only to produce high-quality masks, can also increase instances category classification ability.

3 Information-Enhanced Mask R-CNN

In response to the above issues of Mask R-CNN, we focused on the classic Mask R-CNN and proposed the Information-enhanced Mask R-CNN, called IEMask R-CNN. Compared with the existing Mask R-CNN series models, the main contributions of IEMask R-CNN are generalized as follows: Mask R-CNN, as a classic instance segmentation model, has been proven to be a feasible method. However, since the FPN structure lack useful channel information, global information and low-level texture information, and mask branch cannot obtain useful local-global information, Mask R-CNN is prevented from obtaining high-quality masks and high-accuracy instance category classification. For solving these issues, IEMask R-CNN, is proposed in this paper. In this section, we introduced the overall architecture and each new module of the proposed IEMask R-CNN, as well as analyzed the reasons for their design in detail.

3.1 Overall Architecture

The structure of IEMask R-CNN we designed is illustrated in Fig. 2. We have made corresponding improvements to the structural shortcomings of Mask R-CNN. First, we designed the information-enhanced FPN shown in Fig. 2a, including the channel enhancement module and the global

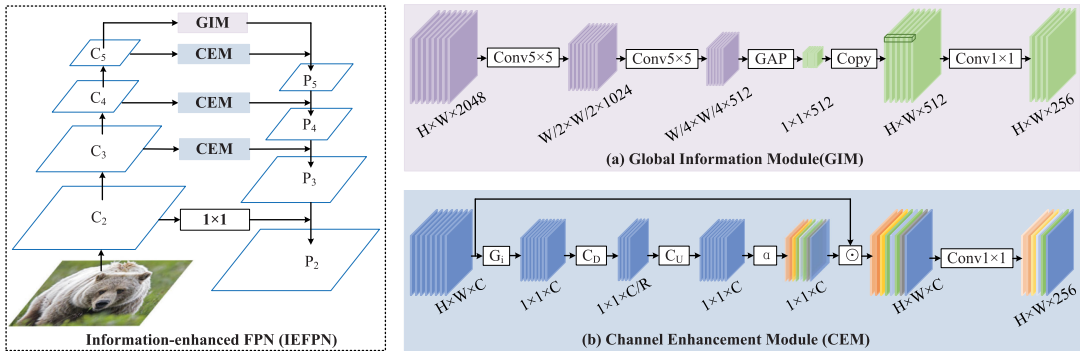

Fig. 3. Illustration of our Information-enhanced FPN (IEFPN). (a) Channel Enhancement Module (CEM), R=16. (b) Global Information Module (GIM).

information module, which can not only make full use of useful channel information but also increase global information to improve the downstream performance of the classification task. Second, we designed the bottom-up path enhancement with adaptive feature fusion as shown in Fig. 2b to effectively propagate the low-level information in the low-level feature map to the high-level feature map to improve the detection effect of large objects. Finally, we designed the encoding-decoding mask head shown in Fig. 2d to replace the mask head in Mask R-CNN, so that the generated mask has stronger local-global information to improve the accuracy of pixel-level classification tasks.

3.2 Information-Enhanced FPN

FPN-based methods adopt 1×1 convolution to reduce channel dimensions of the output feature maps $C_i (i = 2, 3, 4, 5)$ from the backbone, which also loses channel information [25]. In addition, the low-level and high-level information are complementary for object detection, while the semantical information would be diluted in the progress of top-down feature fusion [34]. The FPN structure of Mask R-CNN has been proved to be effective for feature information extraction, but the FPN structure still suffers from loss of useful channel information and lack of global information. Therefore, it is necessary to design an information-enhanced FPN structure to make full use of useful channel information and global information to improve the performance of instance segmentation.

Based on the above analysis, we designed the information-enhanced FPN (IEFPN) as shown in Fig. 3. The output of the original FPN structure can be described as

$$P_i = Conv_{1,1}(C_i) \qquad i = 2, 3, 4, 5, \tag{1}$$

where $Conv_{1,1}(\cdot)$ means a 1×1 convolution, C_i represents the feature level generated by backbone, such as ResNet, and $P_i (i = 2, 3, 4, 5)$ represents the feature level generated by FPN. Here, the function of 1×1 convolution is mainly to reduce the dimensionality to reduce the amount of calculation of downstream tasks. However, the operation of our IEFPN on the feature map C_i generated by the backbone is different from that of FPN. We generated P_i by:

$$P_i = \begin{cases} Conv_{1,1}(C_i) & i = 2 \\ CEM(C_i) & i = 3, 4, \\ CEM(C_i) + GIM(C_i) & i = 5 \end{cases} \tag{2}$$

where $CEM(\cdot)$ and $GIM(\cdot)$ mean channel enhancement function and global information function respectively.

We designed the channel enhancement module shown in Fig. 3b to implement the $CEM(\cdot)$ function in Eq. (2). One of the key points of the channel enhancement module is how to generate different weights for the functions of each channel so that useful channel information can be enhanced and useless information can be weakened, thereby offsetting the loss of useful channel information when compressing channels. As investigated in [35], we designed a self-gating mechanism to capture the channel dependency from the descriptor retrieved by the global average pooling, and then obtain the weight score of the importance of each channel. Therefore, we first employed global average pooling to express the statistics denoting the whole feature map. The global average pooling will reduce the size from $H \times W \times C$ to $1 \times 1 \times C$ as

$$G_i = \frac{1}{H \times W} \sum_{k=1}^{H} \sum_{q=1}^{W} C_i(k, q), \tag{3}$$

where $C_i(k, q)$ is the feature value at position (k, q) in the feature maps. According to [35], we used ReLU and sigmoid functions to realize the gate control mechanism. Let us consider that δ and α are ReLU and sigmoid exercise agents, respectively. Then the gating mechanism is

$$F_s = \alpha(C_U(\delta(C_D(G_i)))), \tag{4}$$

where $C_D(\cdot)$ and $C_U(\cdot)$ mean the channel reduction and channel increase operation, $F_s(\cdot)$ means score matrix. In addition, to obtain the final feature map after channel enhancement, the channel score matrix is multiplied with the original feature map, and finally the feature map is reduced to 256 dimensions through 1×1 convolution. The final feature map output is described as

$$P_i = Conv_{1,1}(F_s \times C_i). \tag{5}$$

We designed the global enhancement module shown in Fig. 3a to implement the $GIM(\cdot)$ function in Eq. (2). The focus of the global information module is how to obtain the global information of the feature map. According to Eq. (2), GIM is only effective for C_5, so we first performed the following operations on C_5:

$$\tilde{C}_5 = Conv^2_{5,2}(C_5), \tag{6}$$

where $Conv^2_{5,2}(\cdot)$ means two consecutive 5×5 convolution with stride 2. We then employ global average pooling to express the statistics denoting the whole feature map.

$$\tilde{G_5} = \frac{1}{H \times W} \sum_{k=1}^{H} \sum_{q=1}^{W} \tilde{C_5}(k, q), \tag{7}$$

where $\tilde{C}(k, q)$ is the feature value at position (k, q) in the feature maps. Finally, we operate on the obtained global information $\tilde{G_5}$ to obtain a global information matrix with the same size as $CEM(C_5)$. Then $GIM(C_5)$ in Eq. (2) can be described as

$$GIM(C_5) = Conv_{1,1}(Copy(\tilde{G_5})), \tag{8}$$

where $Copy(\cdot)$ means copy $\tilde{G_5}$ to $H \times W$, where $H \times W$ is the height and width of C_5.

3.3 Bottom-Up Path Enhancement

The neurons in high layers strongly respond to entire objects while the others are more likely to be activated by local texture has been shown in [36]. However, because FPN has a long path from low-level to topmost layers, high-level feature maps cannot fully use the detailed texture information of low-level feature maps for positioning. Therefore, based on the FPN structure, it is necessary to add a bottom-up path enhancement. In spite of PANet [26] added a path augmentation to Mask R-CNN, because it directly connects the bottom-level information with the top-level information, it ignores the large semantic gap between the bottom-level and top-level feature maps. In addition, it does not solve the weight of the two feature maps in the process of horizontal connection of different features for fusion, which easily causes information redundancy.

We proposed the bottom-up path enhancement of IEMask R-CNN that assigns a learnable weight matrix to each feature map during the feature fusion process. Since the texture information of low-level feature maps gradually spread to high-level feature maps, which not only solves the lack of accurate positioning information in high-level feature maps but also solves the issues of information redundancy and feature importance calibration in the process of feature map fusion. We followed FPN to create layers that produce the same-sized feature maps are in the same network stage. Each feature level corresponds to one stage. We took ResNet and ResNext, etc. as the backbone and used $P_i(i = 2, 3, 4, 5)$ to represent the feature level generated by IEFPN. As shown in Fig. 2b, the output of each level in the bottom-up path enhancement can be described as:

$$N_i = \begin{cases} Conv_{3,1}(P_2) & i = 2 \\ AFF(N_{i-1}, P_i) & i = 3, 4, 5 \end{cases}, \tag{9}$$

where $Conv_{3,1}(\cdot)$ means a 3×3 convolution with stride 1. $P_i(i = 2, 3, 4, 5)$ represents the feature level generated by IEFPN. N_{i-1} is the output of the previous level in the bottom-up path enhancement. $AFF(\cdot)$ means the adaptive feature fusion, which fuses two feature maps P_i and N_{i-1} described as:

$$AFF(N_{i-1}, P_i) = W_1 P_i + W_2 N_{i-1}, \tag{10}$$

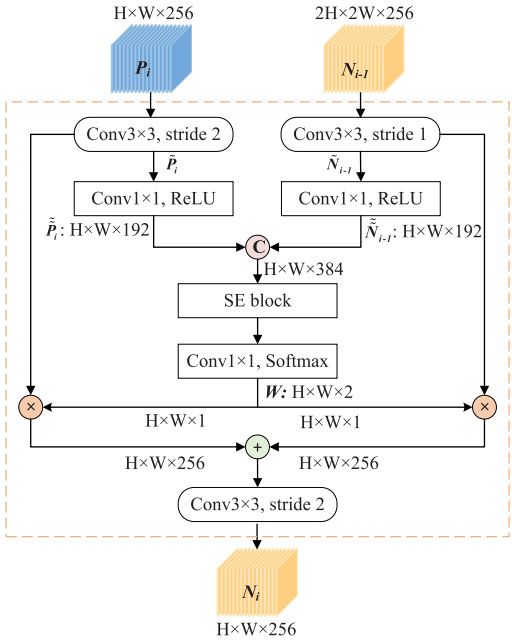

Fig. 4. Adaptive feature fusion(AFF).

where W_1 and W_2 are learnable weight matrixs. In addition, W_1 and W_2 meet the following criteria:

$$\begin{cases} W_1 + W_2 = 1 \\ W_1, W_2 \in [0, 1] \end{cases}, \tag{11}$$

Therefore, the key point is how to obtain the learnable weights W_1 and W_2 in $AFF(\cdot)$.

To make the adaptive fusion features meet Eqs. (10) and (11), we designed the adaptive feature fusion module as shown in Fig. 4. Because N_{i-1} and P_i are two feature maps with different sizes, they are first preprocessed by:

$$\begin{cases} \tilde{N}_{i-1} = Conv_{3,2}(N_{i-1}) \\ \tilde{P}_i = Conv_{3,1}(P_i) \end{cases}, \tag{12}$$

where $Conv_{3,1}(\cdot)$ denotes a 3×3 convolution with stride 1, and $Conv_{3,2}(\cdot)$ denotes a 3×3 convolution with stride 2. Then, to aggregate the channel information of the feature maps, we reduced the number of channels and the amount of subsequent learnable parameters. $\tilde{P}_i$ and $\tilde{N}_{i-1}$ were operated by:

$$\begin{cases} \tilde{\tilde{P}}_i = ReLU(Conv_{1,1}(\tilde{P}_i)) \\ \tilde{\tilde{N}}_{i-1} = ReLU(Conv_{1,1}(\tilde{N}_{i-1})) \end{cases}, \tag{13}$$

where $Conv_{1,1}(\cdot)$ means 1×1 convolution, and $ReLU(\cdot)$ means ReLU activation function.

In order to satisfy Eq. (11), we performed the following operations on the obtained $\tilde{\tilde{P}}_i$ and $\tilde{\tilde{N}}_{i-1}$ to get the final weight matrix:

$$W = soft[Conv_{1,1}(SE(\tilde{\tilde{P}}_i \Theta \tilde{\tilde{N}}_{i-1}))], \tag{14}$$

where Θ and $soft(\cdot)$ mean the connection operation on the channel and softmax function respectively. We used the softmax operation to achieve the restriction condition of Eq. (11). It should be emphasized that W is a three-

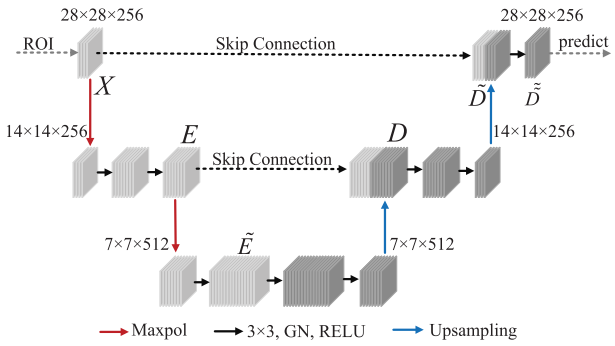

Fig. 5. Encoding-decoding mask head.

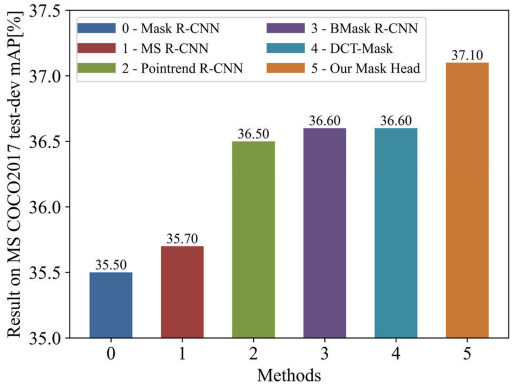

Fig. 6. Mask R-CNN using encoding-decoding mask head comparisons with SOTA methods.

dimensional matrix, which contains weight matrices W_1 and W_2:

$$\begin{cases} W_1 = W_{dim=1} \\ W_2 = W_{dim=2} \end{cases}. \tag{15}$$

3.4 Encoding-Decoding Mask Head

In Mask R-CNN, the mask head performs four continuous 3×3 convolutions on the 14×14 size feature map obtained after RolAlign processing and then upsamples the feature map through deconvolution to obtain a 28×28 feature map, which is used to generate the final mask. He *et al.* [22] has viewed that the mask head in Mask R-CNN is very simple, and it can be improved. Through experimental analysis, Zhao *et al.* [27] found that many errors in the segmentation task are partially or wholly related to contextual relationships and global information for different receptive fields. Therefore, it is necessary to design a mask branch to obtain useful multi-scale local-global information.

Based on the above analysis, the motivation of our work is to gather useful local-global information on the mask branch. The encoding-decoding network with skip connections has been proven to be an efficient network that can effectively obtain local-global information. Therefore, we designed an encoding-decoding mask head in this paper. The architecture of the encoding-decoding mask head is illustrated in Fig. 5. It consists of an encoder (left side) and a decoder (right side). The encoder follows the typical architecture of a convolutional network. The encoder first performs the following operations on the feature map X obtained through ROIAlign processing:

$$E = Conv^2_{3,1}(Down(X)), \tag{16}$$

where $Down(\cdot)$ means maximum pooling of stride 2, $Con^2_{3,1}(\cdot)$ means two continuous 3×3 convolutions with stride 1, and E means output of first block in encoder. It should be noted that the first 3×3 convolution expands the channel of the feature map X twice. After that, the E processed by

$$\tilde{E} = Conv_{3,1}(Down(E)), \tag{17}$$

where $\tilde{E}$ means the final output of the encoder. $Con_{3,1}(\cdot)$ here expands the channel of the feature map to twice the original. Then our decoder first operates on the output $\tilde{E}$ of the encoder through

$$D = Up(Conv^2_{3,1}(\tilde{E})), \tag{18}$$

where $Up(\cdot)$ means bilinear interpolation operation and the D means output of first block in decoder. The first $Conv_{3,1}(\cdot)$ here reduces the channel of the feature map to twice the original size. After that, the D processed by

$$\tilde{D} = Up(Conv^2_{3,1}(E + D)), \tag{19}$$

where $\tilde{D}$ means output of second block in decoder. The final output of our mask head can be described as

$$\tilde{\tilde{D}} = Up(Conv_{3,1}(X + \tilde{D})). \tag{20}$$

Compared with the original mask head in Mask R-CNN, the proposed encoding-decoding mask head uses the encoding-decoding structure and skips connection to process the feature map to generate the mask. The advantage of this structure is to obtain local-global information to improve the quality of mask generation. Additionly, we used 28×28 resolution RoI features instead of 14×14 resolution RoI features, which can make better use of local texture information. At the same time, since we downsampled the 28×28 feature map from the beginning, the complexity of the network will not increase by using the 28×28 resolution RoI features.

We used ResNet-50-FPN as the backbone for model training on the MS COCO2017 training set and test on MS COCO2017 test-dev. As shown in Fig. 6, without any bells and whistles, by just replacing the original mask head of Mask R-CNN with the proposed encoding-decoding mask head, the Mask R-CNN with the proposed encoding-decoding mask head can produce higher-quality masks than some well-designed mask head schemes [12], [13], [23], [24].

4 Experiments and Analysis

In this section, we evaluated the performance of the IEMask R-CNN in instance segmentation through extensive experiments on different datasets. The experimental settings are briefly described in Section 4.1. The ablation experiments are provided in Section 4.2. The comparisons with some existing methods in terms of subjective visual effect and objective evaluations are presented in Section 4.3. Section 4.4 shows the quality evaluation of IEMask R-CNN. Section 4.5

TABLE 2
Detailed Introduction to the Instance Segementation Datasets

Name	Year	Training	Validation	Test	Category	Metrics
MS COCO2017	2017	118287	5000	40670	80	AP, AP_{50}, AP_{75}, AP^s, AP^m, AP^l, $AR^{max=1}$, $AR^{max=10}$, $AR^{max=100}$, AR^s, AR^m, AR^l
Cityscapes	2016	2975	500	1525	8	AP, AP_{50}
LVIS1.0	2020	100170	19809	19822	1203	AP, AP_{50}, AP_{75}, AP^s, AP^m, AP^l, AP_r, AP_c, AP_f

analyzes the IEMask R-CNN and SOTA methods from different aspects. Code is available at *https://github.com/Fhujinwu/IEMask*.

4.1 Experimental Settings

4.1.1 Datasets

To prove the superiority of the IEMask R-CNN in instance segmentation tasks, we carefully selected three instance segmentation datasets with different characteristics as shown in Table 2 and conducted extensive experiments: MS COCO2017 [37], Cityscapes [38], and LVIS1.0 [39]. MS COCO2017 is one of the most commonly used public instance segmentation dataset, which has 80 categories with instance-level annotations. Cityscapes is a real-world dataset containing 5000 high-quality pixel-level annotated images of driving scenes in urban environment. LVIS1.0 is a long-tail instance segmentation dataset consisting of 1203 categories, having more than 2 million high-quality instance mask annotations.

4.1.2 Evaluation Metrics

The evaluation metrics of the experiment is shown in Table 2. On MS COCO2017 and LVIS1.0, we used standard COCO evaluation metrics which include AP, AP_{50}, AP_{75}, AP^s, AP^m, and AP^l. AP is averaged across 10 IoU thresholds ranging from 0.50 to 0.95 in increments of 0.05. AP_{50} and AP_{75} indicate that AP is computed at a single IoU of 0.50 and 0.75, respectively. There are objects of various sizes on MS COCO2017, which are divided into small, medium, and large objects according to their area. AP^s corresponds to AP for small objects whose areas are less than 32^2, AP^m corresponds to AP for medium objects whose areas are between 32^2 and 96^2, and AP^l corresponds to AP for large objects whose areas are more than 96^2. In addition, we also used the six evaluation metrics of $AR^{max=1}$, $AR^{max=10}$, $AR^{max=100}$, AR^s, AR^m and AR^l on MS COCO2017, and the three evaluation indicators of AP_r, AP_c and AP_f on LVIS1.0 to more comprehensively evaluate the effect of our model. AR is the maximum recall given a fixed number of detections per image, averaged over categories and IoUs. LVIS1.0 bin categories based on how many images they appear in: rare (1-10 images), common (11-100), and frequent (>100), so AP_r, AP_c and AP_f can be used to evaluate the effect of the method in the long-tailed distribution dataset. On Cityscapes, we used mAP and AP_{50} to evaluate 8 categories individually and finally evaluate the average value.

4.1.3 Implementation Details

We adopted Mask R-CNN as our baseline, and first gradually verify the effectiveness of information-enhanced FPN, bottom-up path enhancement with adaptive feature fusion, and encoding-decoding mask head of our IEMask R-CNN. Second, re-implement the SOTA methods and our IEMask R-CNN on different datasets and compare them. Third, we qualitatively analyzed our IEMask R-CNN. The detailed implementation details are as follows.

We implemented the networks by PyTorch [40]. The models ran on the 16 GB memory-sized NVIDIA Tesla V100 GPU with CUDA version 9.2 and CUDNN version 6.0. We used the standard $1\times$ training schedule and multi-scale training from Detectron2 [41] by default and used Imagenet pre-trained network as the backbone. On MS COCO2017, the experiments adopted 720 K iterations, batch size 2 on 1 GPUs, and a base learning rate of 0.0025. The learning rate

TABLE 3
Ablation Experiments Results on MS COCO2017 Val

Index	CEM	GIM	Bottom-up path	AFF	Encoding-decoding	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l
1						35.27	56.45	37.72	16.83	38.18	50.00
2	√					35.97	57.48	38.42	17.73	38.63	51.83
3		√				36.11	58.21	38.71	18.62	38.76	51.35
4	√	√				35.79	57.41	38.25	17.54	38.50	51.58
5			√			35.61	56.51	38.10	16.95	37.95	51.51
6			√	√		35.64	56.48	38.23	16.59	38.17	51.52
7					√	36.85	57.08	39.82	17.51	39.40	53.43
8	√		√	√		36.06	57.53	38.18	17.80	38.68	52.13
9	√	√	√	√		36.27	57.96	38.50	18.05	38.75	52.39
10	√	√			√	37.80	**59.09**	40.85	18.93	**40.63**	53.49
11			√	√	√	37.13	57.45	40.26	17.54	39.46	53.83
12	√		√	√	√	37.77	58.51	40.85	18.75	40.34	54.35
13		√	√	√	√	37.79	59.01	40.62	19.06	40.14	**54.41**
14	√	√	√	√	√	**37.83**	58.71	**40.85**	**19.16**	40.49	53.97

TABLE 4
Experimental Results to Evaluate the Effectiveness of AFF

Version	MS COCO2017		Cityscapes		LVIS1.0	
	AP	Δ	AP	Δ	AP	Δ
w/o AFF	37.80		35.80		24.13	
Ours	**37.83**	**0.03** ↑	**36.24**	**0.44** ↑	**25.61**	**1.48** ↑

was reduced by a factor of 10 at iteration 480 K and 640 K. Multi-scale training was used with shorter side randomly sampled from [640, 800]. The short side was resized to 800 in inference. On Cityscapes, we used 160 K iterations and 0.01 base learning rate, and reduce the learning rate at 140 K, where the batch size is 2 on 1 GPU. The shorter side was randomly sampled from [800,1024] in training and resized to 1024 in inference. On LVIS1.0, we used 1440 K iterations, 0.0025 base learning rate, and reduce a factor of 10 at iteration 960 K and 1280 K. All remaining hyper-parameters were kept the same as the Mask R-CNN implemented in Detectron2.

4.2 Ablation Experiments

We demonstrated the effectiveness of our information-enhanced FPN, bottom-up path enhancement with adaptive feature fusion and encoding-decoding mask head through extensive ablation experiments.

4.2.1 Information-Enhanced FPN

As shown in Table 3, the effect of adding IEFPN to the model is comprehensive, and all metrics have been improved. In addition, we also conducted separate experiments on CEM and GIM in IEFPN to verify the effectiveness of these two modules. We only added the CEM, the effect of the model on various metrics has been greatly improved, especially AP^s and AP^l, which have been improved by about 1%. Because the CEM of our IEMask R-CNN can make up for the loss of useful channel information by enhancing the useful information in the channels. The GIM we designed increases the global information of each feature map by adding high-level global semantic information to the feature map, so that the RPN module, downstream object detection module, and instance segmentation module can use the global information as clues, and ultimately improve the effectiveness of the instance segmentation model. In addition, it can be noted that CEM and GIM using IEFPN alone seem to achieve better results than IEFPN, but the results for index 6, 8 & 9, and indexe 11, 12, 13 & 14 in Table 3 show that IEFPN can produce better results in IEMask R-CNN. To the best of our experience, the occurrence of indexe 2, 3 & 4 is due to instability in the network learning process, so we generally combine IEFPN with other modules to get the best performance out of it.

4.2.2 Bottom-Up Path Enhancement With Adaptive Feature Fusion

Without adding other modules, the bottom-up path enhancement with adaptive feature fusion improves the evaluation metric AP by 0.3%, especially the evaluation metric AP^l by 1.52%. It can be seen from the experiment that the improvement of the large instances is significant. The main reason for this improvement is that the high-level feature map of FPN structure lacks the detailed texture

TABLE 5
Comparison With the SOTA Methods on MS COCO2017 Test-Dev

Method	Year	Backbone	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l	$AR^{max=1}$	$AR^{max=10}$	$AR^{max=100}$	AR^s	AR^m	AR^l
Mask R-CNN	2017	R-50-FPN	35.5	56.9	37.9	19.7	37.6	45.9	30.5	47.3	49.5	30.8	52.5	64.0
PANet	2018		36.5	57.6	39.2	20.2	38.6	47.1	30.9	48.3	50.6	31.3	53.3	65.1
MS R-CNN	2019		35.7	57.1	38.1	19.8	37.7	46.3	30.6	47.4	49.5	30.9	52.5	63.9
Pointrend	2020		36.5	57.2	39.3	20.0	38.5	47.5	31.4	48.8	51.1	31.5	54.1	66.4
BMask R-CNN	2020		36.6	57.2	39.5	19.5	38.7	48.0	31.3	48.7	50.9	30.7	54.0	66.4
DCT-Mask	2021		36.6	56.5	39.7	20.0	38.7	47.2	31.4	49.1	51.5	31.9	54.7	66.4
Mask Transfiner	2022		36.9	57.6	39.7	20.1	39.1	48.0	31.5	48.9	51.1	31.6	54.1	66.3
IEMask R-CNN*(ours)*	-		**38.1**	**59.3**	**41.2**	**21.3**	**40.2**	**49.4**	**31.7**	**49.9**	**52.5**	**33.2**	**55.3**	**67.5**
Mask R-CNN	2017	R-101-FPN	37.1	58.9	39.7	20.4	39.7	48.3	31.3	48.5	50.6	31.2	54.0	65.6
PANet	2018		38.0	59.5	40.9	20.9	40.4	49.4	31.7	49.5	51.7	32.3	54.9	66.7
MS R-CNN	2019		37.2	59.0	39.9	20.6	39.7	48.2	31.4	48.5	50.6	31.2	54.1	65.0
Pointrend	2020		38.5	59.5	41.4	20.8	41.0	50.1	32.3	50.1	52.4	31.9	55.9	68.1
BMask R-CNN	2020		38.4	59.3	41.4	20.5	41.0	50.6	32.3	50.0	52.2	31.3	55.7	68.5
DCT-Mask	2021		38.0	58.5	41.2	21.1	40.4	49.1	32.2	50.3	52.6	32.6	56.0	67.9
Mask Transfiner	2022		38.5	59.7	41.5	20.7	41.2	50.5	32.5	50.1	52.2	31.5	55.8	68.2
IEMask R-CNN*(ours)*	-		**39.3**	**61.1**	**42.4**	**21.9**	**41.8**	**51.2**	**32.4**	**50.7**	**53.2**	**33.4**	**56.4**	**68.7**
Mask R-CNN	2017	X-101-FPN	39.6	62.4	42.6	23.3	42.1	50.7	32.5	50.2	52.3	34.0	55.5	66.6
PANet	2018		40.2	62.7	43.5	23.8	42.9	51.3	32.7	50.9	53.1	34.7	56.2	67.4
MS R-CNN	2019		39.6	62.4	42.7	23.5	42.5	50.4	32.4	50.2	52.3	34.1	55.6	66.3
Pointrend	2020		40.8	62.8	44.0	23.4	43.4	52.6	33.3	51.8	54.1	34.7	57.4	**69.8**
BMask R-CNN	2020		40.6	62.4	43.9	23.0	43.4	52.6	33.2	51.4	53.5	33.7	56.9	69.2
DCT-Mask	2021		40.7	62.0	44.2	24.1	43.5	51.8	33.3	52.0	54.4	35.6	**57.9**	69.0
Mask Transfiner	2022		41.1	63.1	44.5	23.7	**44.0**	53.1	**33.4**	51.7	53.9	34.7	57.4	69.4
IEMask R-CNN*(ours)*	-		**41.4**	**63.9**	**45.0**	**24.9**	43.8	**53.3**	33.3	**52.2**	**54.5**	**35.9**	57.6	69.6

Fig. 7. The visual comparison of results for MS COCO2017 val, and all methods used ResNet-50 with FPN (zoom-in for best view.).

information of the low-level feature map, while bottom-up path enhancement with adaptive feature fusion in IEMask R-CNN can effectively transfer the detailed texture information of the low-level feature map to the high-level feature map, improve the positioning ability of the high-level feature map, and then improve the segmentation effect of the large instance. Furthermore, to fully validate the effectiveness of our AFF, we trained a version of IEMask R-CNN (w/o AFF) and validated it on three datasets. As shown in Table 4, after removing the AFF, while the effect did not change significantly on MS COCO2017, the AP score decreased by 0.44% and 1.48% on Cityscapes and LVIS1.0, respectively. Thus, our AFF is indispensable for IEMask R-CNN.

4.2.3 *Encoding-Decoding Mask Head*

We only used the proposed encoding-decoding mask head to replace the mask head in Mask R-CNN without other modifications. The results produced by the model have a significant improvement in all evaluation metrics. The AP and AP_{50} increased by 1.58% and 0.63%, respectively. In particular, the

TABLE 6
Comparison With the SOTA Methods on Cityscapes (Using ResNet-50 With FPN)

Method	Year	person		rider		car		truck		bus		train		motorcycle		bicycle		average	
		AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
Mask R-CNN	2017	31.5	63.1	26.5	66.1	51.0	75.3	28.1	41.2	51.8	72.3	35.8	60.2	17.2	39.2	19.9	53.3	32.7	58.8
PANet	2018	32.8	64.7	26.4	66.6	52.8	76.8	31.7	43.4	53.8	71.9	38.4	62.1	20.1	45.6	21.8	54.8	34.7	60.7
MS R-CNN	2019	32.0	63.8	25.8	65.4	51.4	75.8	30.7	43.3	53.4	71.3	36.1	55.3	19.1	46.4	20.9	53.9	33.7	59.4
Pointrend	2020	33.1	64.5	28.6	68.1	53.8	77.7	34.5	48.0	56.5	74.0	39.5	**62.7**	20.0	44.4	21.1	53.9	35.9	61.7
BMask R-CNN	2020	34.3	65.1	28.3	66.6	54.2	78.0	31.3	44.0	55.8	**74.2**	38.6	60.0	19.5	41.6	22.6	55.2	35.6	60.6
DCT-Mask	2021	34.3	65.6	27.6	65.2	55.2	79.1	**35.8**	**48.7**	56.0	73.6	39.6	58.5	18.7	45.7	23.2	**61.7**	36.3	61.7
Mask Transfiner	2022	31.0	63.7	24.8	65.2	51.2	76.6	30.2	44.8	53.4	73.0	**39.9**	58.3	16.1	38.9	20.2	53.6	33.3	59.3
IEMask R-CNN*(ours)*	-	**36.5**	**67.7**	**30.5**	**69.4**	**55.9**	**79.7**	31.6	43.0	**57.9**	73.0	34.1	57.9	**21.9**	**48.4**	**25.4**	58.8	**36.7**	**62.3**

segmentation result of large objects was improved the most, with AP^l raised from 50.00% to 53.43%. Compared with SOTA methods, Mask R-CNN using encoding-decoding mask head achieved the best results, as shown in Fig. 6. The important reason for this significant improvement is that our encoding-decoding mask head can obtain local-global information very well, thereby producing higher-quality masks.

4.3 Comparative Experiments and Discussion

For proving the performace of the proposed IEMask R-CNN, we compared it to the SOTA methods that include: Mask R-CNN [22], PANet [26], Mask scoring R-CNN [13], Pointrend [12], BMask R-CNN [23], DCT-Mask [24] and Mask Transfiner [42]. We used ResNet-50-FPN, ResNet-101-FPN and ResNeXt-101-FPN as backbone networks respectively to evaluate the performance of our method.

4.3.1 Experiments on MS COCO2017

The comparative experimental results are shown in Table 5. Compared with Mask R-CNN, the average precision (AP) of our method on ResNet-50-FPN, ResNet-101-FPN and ResNeXt-101-FPN are improved by 2.6%, 2.2% and 1.8%, respectively. With ResNet-101-FPN as the backbone, the performance on the evaluation metrics of AP, AP_{50}, AP_{75}, AP^s, AP^m, and AP^1 are 39.3%, 61.1%, 42.4%, 21.9%, 41.8%, and 51.2%, respectively. In addition, our method still has an advantage compared with the SOTA methods. For example, when ResNet-50-FPN was the backbone, the AP score of our IEMask R-CNN is 1.2% higher than the second-best method, Mask Transfiner. The visualization effect is shown in Fig. 7. Compared with the SOTA methods, our method can not only reduce the error rate of classification but also increase the accuracy of details. For example, as shown in Fig. 7 (a1) to (a8), our method can produce higher quality masks. In addition, as shown in Fig. 7 (b1) to (b8), the previous methods all recognize "dog" as "sheep," but our method can accurately identify the object category.

The results of our IEMask R-CNN has been significantly improved because of the improvement brought by the information-enhanced FPN, bottom-up path enhancement with adaptive feature fusion and encoding-decoding mask head of our method. IEMask R-CNN makes full use of the

TABLE 7
Comparison With the SOTA Methods on LVIS1.0

Method	Year	Backbone	AP	AP_{50}	AP_{75}	AP^s	AP^m	AP^l	AP_r	AP_c	AP_f
Mask R-CNN	2017	R-50-FPN	22.44	34.94	23.79	13.94	30.65	40.59	11.45	21.24	28.63
PANet	2018		23.13	35.38	24.58	14.10	31.20	41.89	12.52	21.52	29.59
MS R-CNN	2019		22.26	34.99	23.50	14.02	30.41	40.45	10.86	21.06	28.61
Pointrend	2020		23.98	36.40	25.34	14.53	32.55	43.65	13.57	22.47	30.24
BMask R-CNN	2020		22.95	34.69	24.18	13.52	31.61	43.32	12.07	21.40	29.47
DCT-Mask	2021		23.42	34.81	24.76	14.59	31.77	41.80	12.18	21.77	30.21
Mask Transfiner	2022		24.00	36.29	25.27	14.17	32.34	43.74	13.11	22.58	30.36
IEMask R-CNN*(ours)*	-		**25.61**	**38.32**	**26.76**	**15.77**	**34.00**	**44.13**	**14.30**	**24.00**	**31.24**
Mask R-CNN	2017	R-101-FPN	24.76	38.02	26.36	15.19	33.36	44.40	15.62	23.44	30.25
PANet	2018		25.19	38.11	26.81	15.90	33.67	44.22	14.61	24.05	31.10
MS R-CNN	2019		24.66	37.93	26.35	15.29	33.01	43.27	14.84	23.55	30.23
Pointrend	2020		25.64	38.45	27.08	15.95	34.84	45.66	14.12	24.61	31.85
BMask R-CNN	2020		25.60	38.24	27.28	15.53	34.46	45.97	16.51	24.03	31.35
DCT-Mask	2021		25.39	36.92	27.15	16.09	34.30	45.03	14.53	24.03	31.69
Mask Transfiner	2022		26.21	39.05	27.82	16.03	35.14	46.53	16.21	24.73	32.25
IEMask R-CNN*(ours)*	-		**27.18**	**40.78**	**29.05**	**16.87**	**36.31**	**46.59**	**18.76**	**25.78**	**32.46**
Mask R-CNN	2017	X-101-FPN	25.73	39.32	27.24	16.76	34.49	44.14	14.58	24.48	32.01
PANet	2018		26.55	39.78	28.42	16.91	35.59	45.78	15.19	25.39	32.85
MS R-CNN	2019		26.03	39.60	27.73	16.65	34.65	44.87	15.41	24.89	31.97
Pointrend	2020		23.91	35.68	25.30	15.59	32.35	43.56	10.05	22.20	31.90
BMask R-CNN	2020		25.85	38.22	27.55	15.51	35.33	46.40	13.11	24.56	32.90
DCT-Mask	2021		26.79	38.55	28.78	17.15	36.33	46.12	15.26	25.51	33.30
Mask Transfiner	2022		27.45	40.66	29.00	17.70	37.00	47.40	14.74	26.59	**33.99**
IEMask R-CNN*(ours)*	-		**27.91**	**41.24**	**29.66**	**17.99**	**37.02**	**47.60**	**17.50**	**26.63**	33.90

channel information in IEFPN through the CEM, and increases the ability of downstream classification tasks through the GIM in IEFPN, enhances the dissemination of detailed texture information of low-level feature maps through the bottom-up path enhancement with adaptive feature fusion to enhance the localization capabilities of high-level feature maps, and obtain local-global information through encoding-decoding mask head to obtain high-quality masks.

4.3.2 Experiments on Cityscapes

As shown in Table 2, Cityscapes and MS COCO2017 are completely different types of datasets. The image resolution is large and it has a high-quality mask, but the number of images and target types are relatively small. It contains 2975 training sets and 500 training sets. In order to more comprehensively evaluate the effects of our IEMask R-CNN and the SOTA methods, we used ResNet-50-FPN as the backbone, and calculated the mAP of the 8 types of instances in Cityscapes and the average value of the mAP of the 8 types of instances.

The experimental results on the realistic and high-resolution dataset of Cityscapes are shown in Table 6. IEMask R-CNN obtained larger improvements on this dataset than on MS COCO2017. Compared with Mask R-CNN, IEMask R-CNN increased from 32.7% and 58.8% to 36.7% and 62.3% on the "average" AP and AP_{50}, respectively. In addition, our method still achieved a higher AP on most instance categories compared to recent methods. This further demonstrates the superiority of our method for high-quality mask prediction. On this relatively small dataset, our method can still obtain high-quality instance segmentation results, which further proves the superiority of our method.

4.3.3 Experiments on LVIS1.0

LVIS1.0 is a new dataset for the segmentation of large-scale vocabulary instances, and it was released in 2020. Although LVIS1.0 uses the pictures on MS COCO2017, the pictures of LVIS1.0 have more detailed annotations and more categories (1203 categories). LVIS1.0 poses a greater challenge to the instance segmentation model, but our IEMask R-CNN still shows the best results. Table 7 shows the results of IEMask R-CNN and the SOTA methods under different backbone networks. Our method IEMask R-CNN surpasses the SOTA methods in all evaluation metrics. Especially compared with Mask R-CNN, our IEMask R-CNN used ResNet-50-FPN as the backbone network, which improved the AP by 3.17%, especially on AP_{50} and AP^l. In addition, when ResNet-101-FPN was used as the backbone network, the evaluation metrics are also significantly improved. Our method showed its superiority in instance segmentation tasks compared with other SOTA methods on LVIS1.0. For example, when using ResNet-50-FPN as the backbone network, the AP score of IEMask R-CNN is 1.61% higher than the second-best method, Mask Transfiner.

For the reason that LVIS1.0 is a dataset with long-tailed distribution characteristics, many current methods do not perform well in categories with a small amount of data, which poses a challenge to the current methods. However, as shown in Table 7, our IEMask R-CNN has improved significantly on AP_r and AP_c, especially with an accuracy of 18.76% on AP^r. It can be seen that our method also has a better effect on the long-tail distribution dataset.

Fig. 8. More example result pairs from Mask R-CNN (left images) $versus$ IEMask R-CNN (right images), using ResNet-50 with FPN (zoom-in for best view).

TABLE 8
Comparison With SOTA Methods on MS COCO2017 Val

Method	Year	AP	Params.(M)	FLOPs(G)	Time(ms/Img)
Mask R-CNN	2017	35.27	**44.3**	203.7	**62.7**
PANet	2018	36.20	68.1	228.1	65.8
YOLACT-550	2019	29.00	140.2	**61.6**	-
Cascade Mask	2019	35.90	112.5	355.3	-
Pointrend	2020	36.14	60.2	214.1	93.7
BMask R-CNN	2020	36.30	47.1	218.6	81.1
DCT-Mask	2021	36.33	96.8	198.8	71.6
SOTR [43]	2021	35.92	63.2	-	145.3
Mask Transfiner	2022	36.49	52.9	739.3	241.1
Ours†	-	37.77	75.0	326.3	82.4
Ours	-	**37.83**	140.0	340.0	90.8

Average precision (AP), parameters (Params.), floating-point operations per second (FLOPs) and inference time (Time). Ours† indicates IEMask R-CNN (w/o GIM).

4.4 Qualitative Results

We provided some visualization results on MS COCO2017 validation set to compare our method with Mask R-CNN and further prove the effectiveness of our method. As shown in Fig. 8, the qualitative results were achieved on the backbone of ResNet-50-FPN. IEMask R-CNN predicted mask with substantially higher quality than Mask R-CNN, especially for the hard regions, such as the wing of the airplane (the first line), the mouth of the zebra (the third line), the legs of the bear (the sixth line). In addition, the instance category classification effect of IEMask was also higher than Mask R-CNN, especially for similar categories, such as the bird (the second line). The qualitative results further prove that compared with the original Mask R-CNN, our IEMask R-CNN can not only product high-quality masks, but also improve the performance of instance category classification.

4.5 Analysis of Methods

We comprehensively evaluated our IEMask R-CNN and SOTA methods using four dimensions: average precision (AP), parameters (Params.), floating-point operations per second (FLOPs) and inference time (Time). As shown in Table 8, the AP score of our IEMask R-CNN is 1.43% higher than the second-best method, Mask Transfiner. Params. and FLOPs are two measures of model size. Our method has a higher number of parameters compared to SOTA methods, but the FLOPs of our method are still kept within an acceptable range. In real-world scenarios, the inference time of the method is particularly important. As shown in Table 8, our method not only achieves high AP score, but also exhibits faster inference times ($versus$ Pointrend, SOTR and Mask Transfiner, etc.).

Having a large number of parameters may be a drawback of the existence of IEMask R-CNN. Therefore, we believe that in the case of constraints on the model parameters, we can use the "IEMask R-CNN (w/o GIM)" version, which can obtain higher AP score than SOTA methods with lower parameters.

5 Conclusion

In this work, we proposed a proposal-based instance segmentation method called IEMask R-CNN for high-quality instance segmentation. IEMask R-CNN makes full use of the channel information and increases the ability of downstream classification through the information-enhanced FPN, and enhances the dissemination of detailed texture information of low-level feature maps through the bottom-up path enhancement with adaptive feature fusion to enhance the localization capabilities of high-level feature maps, and obtain local-global information through encoding-decoding mask head to obtain high-quality masks.

In addition, we have conducted extensive experiments on three different types of instance segmentation datasets: MS COCO2017, Cityscapes, and LIVS1.0. Our IEMask R-CNN has advantages in both visual quality evaluation and objective evaluation metrics, which greatly proves the superiority of our method in the instance segmentation task. We believe IEMask R-CNN can serve as a strong baseline for high-quality instance segmentation.

References

[1] F. Wu *et al.*, "Weakly semi-supervised deep learning for multi-label image annotation," *IEEE Trans. Big Data*, vol. 1, no. 3, pp. 109–122, Sep. 2015.

[2] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep feature learning for medical image analysis with convolutional autoencoder neural network," *IEEE Trans. Big Data*, vol. 7, no. 4, pp. 750–758, Oct. 2021.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[9] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[10] Q. Yan *et al.*, "COVID-19 chest CT image segmentation network by multi-scale fusion and enhancement operations," *IEEE Trans. Big Data*, vol. 7, no. 1, pp. 13–24, Mar. 2021.

[11] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9156–9165.

[12] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9796–9805.

[13] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6402–6411.

[14] J. Pei, H. Tang, C. Liu, and C. Chen, "Salient instance segmentation via subitizing and clustering," *Neurocomputing*, vol. 402, pp. 423–436, 2020.

[15] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2858–2866.

[16] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan, "Proposal-free network for instance-level object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2978–2991, Dec. 2018.

[17] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, "DCAN: Deep contour-aware networks for object instance segmentation from histology images," *Med. Image Anal.*, vol. 36, pp. 135–146, 2017.
[18] D. Zhang, Y. Song, S. Liu, D. Feng, Y. Wang, and W. Cai, "Nuclei instance segmentation with dual contour-enhanced adversarial network," in *Proc. IEEE 15th Int. Symp. Biomed. Imag.*, 2018, pp. 409–412.
[19] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "InstanceCut: From edges to instances with MultiCut," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7322–7331.
[20] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," 2016, *arXiv:1611.05424*.
[21] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 282–298.
[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[23] T. Cheng, X. Wang, L. Huang, and W. Liu, "Boundary-preserving mask R-CNN," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 660–676.
[24] X. Shen *et al.*, "DCT-mask: Discrete cosine transform mask representation for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8716–8725.
[25] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 592–12 601.
[26] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.
[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239.
[28] V. Kulikov, V. Yurchenko, and V. Lempitsky, "Instance segmentation by deep coloring," 2018, *arXiv:1807.10007*.
[29] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, "A dataset and a technique for generalized nuclear segmentation for computational pathology," *IEEE Trans. Med. Imag.*, vol. 36, no. 7, pp. 1550–1560, Jul. 2017.
[30] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6517–6525.
[32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.
[33] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.
[34] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 821–830.
[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[36] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[37] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[38] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.
[39] A. Gupta, P. Dollar, and R. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5351–5359.
[40] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[41] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019.
[42] L. Ke, M. Danelljan, X. Li, Y.-W. Tai, C.-K. Tang, and F. Yu, "Mask transfiner for high-quality instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4412–4421.
[43] R. Guo, D. Niu, L. Qu, and Z. Li, "SOTR: Segmenting objects with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7137–7146.

Xiuli Bi received the BSc and MSc degrees from Shanxi Normal University, China, in 2004 and 2007, respectively, and the PhD degree in computer science from the University of Macau, in 2017. She is currently an associate professor with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China. Her research interests include image processing, multimedia security, and forensic.

Jinwu Hu (Student Member, IEEE) received the BS degree in network engineering from Foshan University, Foshan, China, in 2020. He is currently working toward the postgraduate degree with the Chongqing University of Posts and Telecommunications, Chongqing, China, and will receive the MS degree in the computer technology in 2023. His research interests include image processing and deep learning.

Bin Xiao received the BS and MS degrees in electrical engineering from Shanxi Normal University, Xian, China, in 2004 and 2007, respectively, and the PhD degree in computer science from Xidian University, Xian, China. He is now a professor with the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include image processing and pattern recognition.

Weisheng Li (Member, IEEE) received the BS degree from the School of Electronics and Mechanical Engineering, Xidian University, Xian, China, in July 1997, and the MS and PhD degrees from the School of Electronics and Mechanical Engineering and School of Computer Science and Technology, Xidian University, in July 2000 and July 2004, respectively. Currently, he is a professor with the Chongqing University of Posts and Telecommunications. His research focuses on intelligent information processing and pattern recognition.

Xinbo Gao (Senior Member, IEEE) received the BEng, MSc, and PhD degrees in signal and information processing from Xidian University, Xian, China, in 1994, 1997, and 1999, respectively. From 1997 to 1998, he was a research fellow with the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a postdoctoral research fellow with the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong. He had been with the School of Electronic Engineering, Xidian University Xian, China, from 2001 to 2020. Since 2020, he has been the president with the Chongqing University of Posts and Telecommunications, Chongqing, China. His current research interests include multimedia analysis, computer vision, pattern recognition, machine learning, and wireless communications. He has published five books and approximately 200 technical articles in refereed journals and proceedings, including the *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Systems, Man, and Cybernetics*, *International Journal of Computer Vision*, and *Pattern Recognition* in the above areas. He is a fellow of the Institution of Engineering and Technology. He has served as the general chair/co-chair, the Program Committee chair/co-chair, or a PC member for approximately 30 major international conferences. He is on the editorial boards of several journals, including the *Signal Processing* (Elsevier) and *Neurocomputing* (Elsevier).