J Inf Process Syst, Vol.18, No.3, pp.411~427, June 2022
https://doi.org/10.3745/JIPS.02.0174

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

SEL-RefineMask: A Seal Segmentation and Recognition Neural Network with SEL-FPN

Ze-dong Dun, Jian-yu Chen, Mei-xia Qu, and Bin Jiang*

Abstract

Digging historical and cultural information from seals in ancient books is of great significance. However, ancient Chinese seal samples are scarce and carving methods are diverse, and traditional digital image processing methods based on greyscale have difficulty achieving superior segmentation and recognition performance. Recently, some deep learning algorithms have been proposed to address this problem; however, current neural networks are difficult to train owing to the lack of datasets. To solve the afore-mentioned problems, we proposed an SEL-RefineMask which combines selector of feature pyramid network (SEL-FPN) with RefineMask to segment and recognize seals. We designed an SEL-FPN to intelligently select a specific layer which represents different scales in the FPN and reduces the number of anchor frames. We performed experiments on some instance segmentation networks as the baseline method, and the top-1 segmentation result of 64.93% is 5.73% higher than that of humans. The top-1 result of the SEL-RefineMask network reached 67.96% which surpassed the baseline results. After segmentation, a vision transformer was used to recognize the segmentation output, and the accuracy reached 91%. Furthermore, a dataset of seals in ancient Chinese books (SACB) for segmentation and small seal font (SSF) for recognition were established which are publicly available on the website.

Keywords

Character Recognition, Feature Extraction, FPN, RefineMask, Seal Character Segmentation

1. Introduction

As one of the carriers of information, seals in ancient Chinese books (SACB) evolved synchronously with human civilization development for their precious semantic concentration [1]. Generally, the seal is printed on the cover or last page of books, reflecting the trajectory of ancient books, which is an important basis for academic research and version identification. By studying ancient seals, we not only obtained information about the collectors of ancient books, but also analyzed the historical heritage of ancient books. Therefore, automated ancient seal segmentation and recognition are extremely valuable for ancient document research and auxiliary reading of ancient texts. However, seal segmentation and recognition in ancient Chinese books are more challenging than general scene text recognition (STR) [2], which can be summarized as follows. (1) There are different Chinese chirographic fonts in different periods, and every chirographic font has different writing styles, which increases intra-class variation and

Manuscript received December 16, 2021; first revision March 10, 2022; accepted March 15, 2022.
***Corresponding Author:** Bin Jiang (jiangbin@sdu.edu.cn)
School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China ({dunzedong, cjyyy}@mail.sdu.edu.cn, {mxqu, jiangbin}@sdu.edu.cn)

inter-class similarity with complicated Chinese character structures. (2) Clutter, occlusion, adhesion, and omitted strokes existed in ancient seals. (3) Owing to the diversity of seal-cutting techniques, the forms and styles of seals also differ, making them harder to segment and recognize.

Currently, the mature technique of computer vision, known as scene text recognition, recognizing text in natural scenes is usually considered a special form of optical character recognition (OCR) [3]. Although OCR performs well in scanned documents, STR remains challenging owing to various fonts, complex backgrounds, and imperfect imaging conditions. There are two popular STR categories: segmentation-based and segmentation-free. Segmentation-based methods [4] typically involve three steps: image pre-processing, character segmentation, and character recognition. Segmentation-based methods first locate the position of each character and then apply a character classifier to recognize each character. Segmentation-free methods [5] recognize the text line as a whole and focus on transforming the entire text image into a character sequence directly by an encoder-decoder framework which is effective when applied to scenes with fewer types of characters.

The seal image can also be digitally recognized as a specialized scene. Owing to the lack of related open datasets on Chinese ancient seal samples and language barriers, there is little research on seal segmentation and recognition in ancient Chinese books using computer vision methods. Moreover, owing to the variety of characters in the seal, we adopted segmentation-based methods to segment the seal first and then recognize each character. Datasets SACB and SSF were created for training and testing in these two stages. Contributions of our work:

(1) We implemented a baseline experiment on outstanding instance segmentation networks. We also designed a selector SEL-FPN (an essential component in SEL-RefineMask) which can intelligently select a specific layer representing different scales in FPN. Compared to the baseline results, the top-1 accuracy increased by 3.03%.

(2) We established SACB and SSF datasets. The SACB dataset contains 12,000 seal images which are collected from digital platforms such as Jangseogak of the Academy of Korean Studies and Kyujanggak Institute for Korean Studies and is used in segmentation. The SSF dataset contains 1,000 small seal-font characters used for recognition.

The paper proceeds as follows: in Section 2, we introduce the current status of research in the field of character segmentation and recognition, including digital image processing and deep learning methods. Subsequently, we analyzed the main difficulties and solutions of seal segmentation and recognition. Section 3 introduces the establishment of SACB and SSF datasets. The SEL-RefineMask framework was also used to segment seals, and the vision transformer (ViT) module was used to recognize characters. The main processes of segmentation and recognition are also introduced. In Section 4, we compared our proposed method with other baseline methods to verify its reliability. Finally, in Section 5, we briefly summarize the achievements and propose future work in the seal research field.

2. Related Works

In early research, character features were used for text segmentation and recognition, such as histograms of oriented gradients (HOG), connected components, and stroke width transform. However, the performance of these methods is not outstanding because of their low-capacity features. With the

development of deep learning, various methods have been proposed and substantial advancements have been made in innovation, utility, and efficiency.

2.1 Digital Image Processing Methods for Character Segmentation

Most conventional character segmentation algorithms are based on the similarity of grey-scale values, while the other algorithms are based on abrupt changes. Character outline segmentation is based on the premise that the boundaries of different characters completely vary from each other and from the background in an image. Other segmentation algorithms are based on a set of predefined criteria, such as threshold, region growing, splitting, and aggregation.

Otsu [6] proposed a nonparametric, unsupervised threshold selection method from grey-level histograms, that is, the projection method. An optimal threshold value was selected according to the discriminant criterion to maximize the grey-scale separability of the results. This process is relatively simple, and the segmentation task can be accomplished using the zero- and first-order cumulative moments of the grey-scale histogram. Jiang et al. [7] proposed an optimal two-scan concatenated labelling algorithm based on Otsu [6], which solves the problem that the projection method cannot segment inclined and interlaced characters. However, adjacent characters with adhering strokes are usually treated only as a single character using this method. Xu et al. [8] proposed a contour and skeleton analysis-based method for separating characters using adhering strokes, which highlights the relevance of character separation points. However, segmenting characters with high adhesion remains difficult. Ma and Yang [9] developed the traditional drip method to segment a long string of sticky characters. The outline information of the Chinese characters plays a crucial role in this modified drip algorithm to select the starting point. Hu et al. [10] proposed a new automatic seal imprint verification approach. Owing to the efficient and accurate registration method, set of effective features, and elaborate implementations, this approach satisfies most practical requirements. Li et al. [11] proposed a character segmentation method to predict the candidate character area without any labelled training data containing character coordinate information. A retrieval-based recognition system that focuses on a single character was also proposed to support seal retrieval and matching.

2.2 Deep Learning Methods for Character Segmentation

Most current methods for character segmentation are based on deep learning algorithms. These methods are trained to generate text proposal regions first using a text detection framework and then recognizing them with a separate text recognition model.

Jaderberg et al. [12] first generated holistic text proposal regions using an ensemble framework and then used a classifier for recognition. Lyu et al. [13] proposed a framework based on Mask R-CNN [14] to detect curved texts. Lyu's framework can recognize 26 letters and 10 numbers with superior recall for curved text and supports overall end-to-end training. However, it is more suitable for simple English recognition scenarios and limited for variant Chinese text because it can only recognize 36 characters. Li et al. [15] proposed an end-to-end trainable framework that can recognize texts of any shape. Significant breakthroughs have been made in the reading of irregularly shaped scenarios. The region of interest (RoI) mask is generated to provide text features for the recognizer, which is an LSTM-decoder inspired by the attention mechanism of Bahdanau-style [16] and can sequentially select relevant character features for

decoding. Lamb et al. [17] proposed an end-to-end regularized residual U-Net called KuroNet to automatically recognize Kuzushiji historical texts and transcribe them into modern Japanese characters. Liu et al. [18] proposed a new integrated model for end-to-end fast text detection and recognition, called FOTS, in which RoIRotate was proposed. Similar to both the RoIPool and RoIAlign methods, the main purpose of this operation is to pool text regions with direction. RoIRotate enables the end-to-end concatenation of text detection and recognition. FOTS has achieved state-of-the-art results on the publicly available ICDAR 2015 [19], ICDAR 2017 MLT [20], and ICDAR 2013 datasets. In the field of image classification, Aamir [21] proposed an optimized architecture for image classification using a convolutional neural network.

The application of these scene text recognition algorithms to seal segmentation and recognition is limited for two reasons. (1) There are no ready-made seal datasets. (2) The design of these deep learning networks is not applicable for scenarios with many types of characters. Using segmentation-based methods, we decoupled segmentation and recognition; first we segmented the seal and then recognized the segmentation results. Subsequently, datasets SACB and SSF were established.

3. Data and Methods

3.1 Dataset Creation

Open-source OCR datasets are based on English alphabets, digits, or simplified Chinese characters, such as the ICDAR 2013 dataset which labels 462 images in English, ICDAR 2015 dataset which labels 1,000 images in English, ICDAR 2017 mixed language dataset which labels more than 12,263 images, and ICDAR 2019 MLT dataset which labels 20,000 images. In recent years, several Chinese text datasets have been published, such as CTW-18 which contains more than 1 million Chinese character instances with 3,850 character-categories and CASIA-10K which contains 10,000 images under various scenarios. However, ancient seal images and character annotations are scarce. No dataset with image annotation is available for the study of ancient seal images. In this study, we built a semantic annotation-level ancient seal image dataset called SACB. First, we collected numerous seal images as the initial data source. Seal images that satisfy these requirements were selected through well-established digital platforms in Northeast Asia, such as Digital Jangseogak and Kyujanggak. The collected seal data encompassed various background conditions, and the seal data had different resolutions and qualities and contained different numbers of characters, in which 12,000 images were selected as the original images, which ensured the richness of the dataset. The characters in the seals were segmented using an instance segmentation algorithm, and each of them was considered an instance. A character instance required pixel-level annotation. In this study, LabelMe [22] was used to annotate the seal dataset. Dataset link at https://gitee.com/vaco/sealdata.

Creation of ancient seal dataset

According to experimental requirements, this study follows the annotation format of COCO2017 public dataset [23] to annotate the collected images. The main object annotated in this study was small seal character instances. We used LabelMe for the annotation, which automatically generates the corresponding JSON format annotation file, and the image data annotation is shown in Fig. 1.

Fig. 1. Original seal and annotated seal images.

In this study, we collected different shapes of seals. The number of characters contained in the seal ranges from 1 to 12 or more, and rarely exceeds 12 characters. The SACB contains 12,000 seal samples, which are evenly distributed according to the number of characters in the seal. There were 600 seal samples with more than 12 characters and 950 seal samples for each class with characters ranging from 1 to 12. The shape of the seal includes circles, squares, rectangles, and other evenly distributed shapes. Seal styles and cutting brushworks are rich and diverse during different periods.

Creation of ancient SSF dataset

After segmentation, the segmented characters from the seal must be recognized. The characters in ancient seals are usually SSF which are popular in ancient Chinese seal carvings, as shown in Fig. 2. There is no SSF dataset available yet; therefore, in this study, we created an SSF dataset, containing 1,000 categories of small seal script character font, and each character category contains 10 pictures. The SSF dataset had 10,000 pictures.

Fig. 2. Small seal font (SSF) dataset.

3.2 Algorithm Design

Characters in antique seals are difficult to segment because of their dense alignment; thus, traditional instance segmentation networks cannot provide accurate results. We improved the RefineMask network and proposed a neural network called SEL-RefineMask. We added an SEL-FPN layer as an improvement, which intelligently selects the feature maps in the FPN. Different feature maps have different representation capabilities for seals with different character densities. The SEL-RefineMask can perform instance segmentation of characters in the seal and subsequently conduct the recognition of characters. The SEL-RefineMask consisted of three parts: a convolutional layer for extracting features, character segmentation branch, and character recognition branch.

3.2.1 Architecture

The framework of the SEL-RefineMask is shown in Fig. 3. The character segmentation branch extracts

features using the backbone network ResNet-50 [24]. Inspired by the FPN [25], we connected the low-level feature map with the high-level semantic feature map and proposed an SEL-FPN module. The SEL-FPN can intelligently select relevant layers to generate anchor frames. Then, the semantic [26] and mask [26] heads were combined to use the semantic fusion module (SFM) [26] and boundary-aware refinement (BAR) [26] for high-quality segmentation. The segmentation branch performs a pixel-by-pixel analysis of the characters such that a more precise position of each character can be detected.

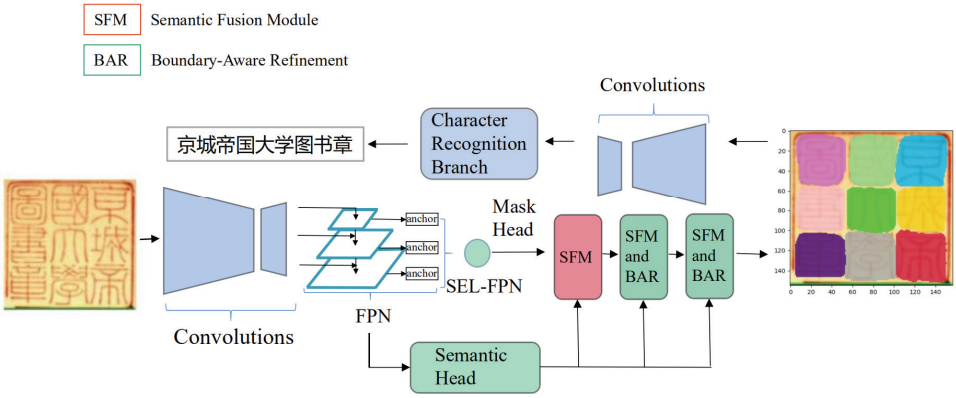


Fig. 3. Framework of SEL-RefineMask.

After segmentation, we obtained the precise position of each character in the seal. ViT [27] and other classification networks were trained on self-built datasets (SSF) to classify the small seal character images generated in the segmentation stage and convert them into simplified Chinese characters.

3.2.2 Segmentation branch

Mask R-CNN, a two-stage instance segmentation network, has superior performance. However, the semantic masks are still very coarse because of the downsampling operation in both the feature pyramid [27] and instance-wise pooling process, particularly for large objects. RefineMask performs excellently for high-quality segmentation of instances and incorporates fine-grained features during the instance-wise segmenting process in a multi-stage process. Therefore, it is more effective for the segmentation of sticky characters than traditional methods. We considered each character as an instance and used the modified SEL-RefineMask as a character segmenter.

After extracting the features by the backbone network, we applied the SEL-FPN to select the corresponding layers and generate the candidate regions of characters by the region proposal network (RPN) [28]. The RPN loss function is defined as follows:

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

In Eq. (1), $\boldsymbol{i}$ is the anchor [28] frame index, and p_i is the probability of being a positive sample. p_i^* is the prediction probability of the corresponding ground-truth (GT).

The ratio of the positive to negative samples was approximately 1:3. t is the vector of the four parameterised coordinates of the predicted bounding box, and t^* is the true bounding box of the positive

sample. The classification loss L_{cls} is the log loss over two categories, that is whether there are objects or not. For the regression loss, we use $L_{reg}(t_i\ ,\ t_i^*) = R(t_i - t_i^*)$, where R represents the robust smooth L1 loss defined in Fast R-CNN [29]. $p_i^* L_{reg}$ represents the regression loss for the anchor of the positive sample, whereas the others are ignored. The outputs of the cls and reg branches consist of $\{p_i\}$ and $\{t_i\}$, respectively. L_{cls} and L_{reg} are normalised by N_{cls} and N_{reg} and then weighted by the balance factor λ.

SEL-RefineMask relies on two small network modules, the semantic and mask heads, to perform high-quality instance segmentation. The semantic head uses the P_2 (the highest-resolution feature map in FPN) as the input and performs semantic segmentation. The fine-grained features generated by the semantic head are utilized to facilitate multistage (three-stage) instance segmentation in the mask head. At each stage, the mask head incorporates the semantic features and semantic mask extracted from the fine-grained features, thereby increasing the spatial size of the features for a finer instance mask prediction. Each stage has an SFM module to fuse the instance features and instance mask obtained from the previous stage, semantic features, and semantic mask generated by the semantic head. In addition, a BAR strategy was used in the mask head to explicitly focus on the boundary regions for predicting crisp boundaries. The loss function $\mathcal{L}^k$ for the k-th stage (k = 2, 3) with an output size of $S_k \times S_k$ is defined as follows:

$$\mathcal{L}^k = \frac{1}{\delta_n} \sum_{n=0}^{N-1} \sum_{i=0}^{S_k-1} \sum_{j=0}^{S_k-1} R_{nij}^k \cdot l_{nij} \tag{2}$$

where N is the number of instances and l_{nij} is a binary cross-entropy loss at position (i, j) for instance n. R^k represents the regions determined by both the GT mask and predicted mask of its preceding stage. B^k represents the binary instance mask of stage k which is defined as follows:

$$B^k(i,j) = \begin{cases} 1, & if\ d_{ij} \le \hat{d} \\ 0, & otherwise \end{cases} \tag{3}$$

B^k is defined as the region consisting of pixels whose distance to the mask contour is less than $\hat{d}$ pixels. R^k is defined as follows:

$$R^k = f_{up}(B_G^{k-1} \vee B_p^{k-1}) \tag{4}$$

$$\delta_n = \sum_{n=0}^{N-1} \sum_{i=0}^{S_k-1} \sum_{j=0}^{S_k-1} R_{nij} \tag{5}$$

where f_{up} denotes the bilinear upsampling operation with a scale factor of 2, B_G^{k-1}denotes the boundary region of the GT mask in stage $k-1$, B_p^{k-1} denotes the boundary region of the predicted mask in stage $k-1$, and the two boundary regions perform the union operations.

As shown in Fig. 4, SEL-RefineMask first generates bounding boxes which surround the characters by regression. Then, the corresponding masks are obtained by the deconvolution network in the mask branch, where 0 indicates that the pixel at that position is not related to the character, and 1 indicates that the pixel is in the character. The outlines of the characters inside the border can be preserved using these masks. Meanwhile, it is possible to remove the residual pixels that may exist in the bounding box caused by strokes of the seal border and adjacent characters. For example, in the character shown in Fig. 4, there are few residual pixels in the bounding box caused by the seal border on the upper left and the adjacent

character strokes on the lower right, which affect the accuracy of the subsequent classification. Only the character instances in the regression bounding boxes were retained to remove the influence of the seal border or adjacent character strokes by training the mask branch. The pixels in the characters are retained, and the remaining pixels are set to zero after the masking operations aforementioned. After binarizing the character image, a greyscale image was obtained and finally fed into the convolutional network for further classification and recognition.

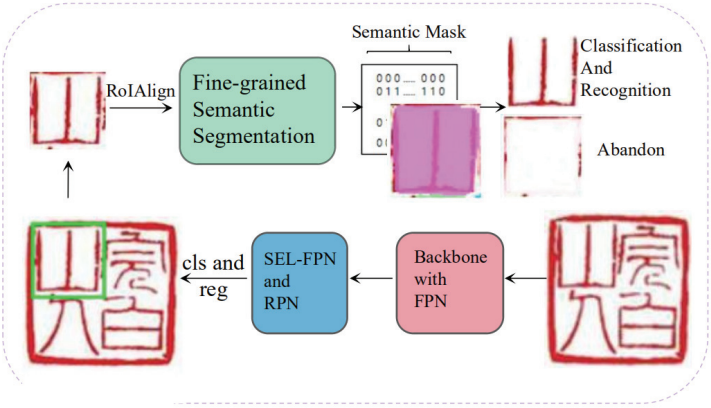


Fig. 4. Overview of segmentation framework.

3.2.3 Recognition branch

An image of each character in the seal was obtained after segmentation. The SSF script library contained 1,000 small seal script fonts. We used ViT to classify the characters generated in the segmentation stage. We split a character image into fixed-size patches, linearly embedded each of them, added position embedding, and fed the resulting sequence of vectors into a standard transformer encoder. To perform the classification, we used the standard approach of adding an extra learnable classification token to the sequence.

As shown in Fig. 5, the 1D sequence of the token embedding was used as the standard input of the transformer. To process the 2D images, we first split the character image $x \in \mathbb{R}^{H\times W\times C}$ into a sequence of fixed-size 2D patches $x_q \in \mathbb{R}^{\gamma\times(Q^2\cdot C)}$, where (H, W) is the resolution of the original image, C is the number of channels, γ is the number of patches, and (Q, Q) is the resolution of each image patch. We flattened the patches $(p_1, p_2, \dots p_n)$ and mapped them to D dimensions $(x_1, x_2, \dots x_n)$ with a trainable linear projection $(z_1, z_2, \dots z_n)$. We then added positional encoding vectors to $(z_1, z_2, \dots z_n)$. Similar to BERT's class token, a learnable embedding $(z_0 = x_{class})$ is added to the sequence of embedded patches, whose output c_0 of the transformer encoder is used for image classification. $(z_0, z_1, z_2, \dots z_n)$ are entered into the transformer encoder network to generate $(c_0, c_1, c_2, \dots c_n)$, where $(c_1, c_2, \dots c_n)$ can be omitted. The softmax classifier received c_0 as the input. Finally, we obtained the class of characters.

In the training, we first randomly initialized the ViT network parameters and then pre-trained them. Subsequently, ViT is fine-tuned on the SSF training set. Thus, the training process is complete. Then, we tested ViT on the SSF testing set to obtain the accuracy. We used SVM, Alex Net, VGG16, ResNet-50, ResNet-101, ResNeSt-50, and ResNeSt-101 for the comparison experiments. ViT showed superior performance over the other models in the classification

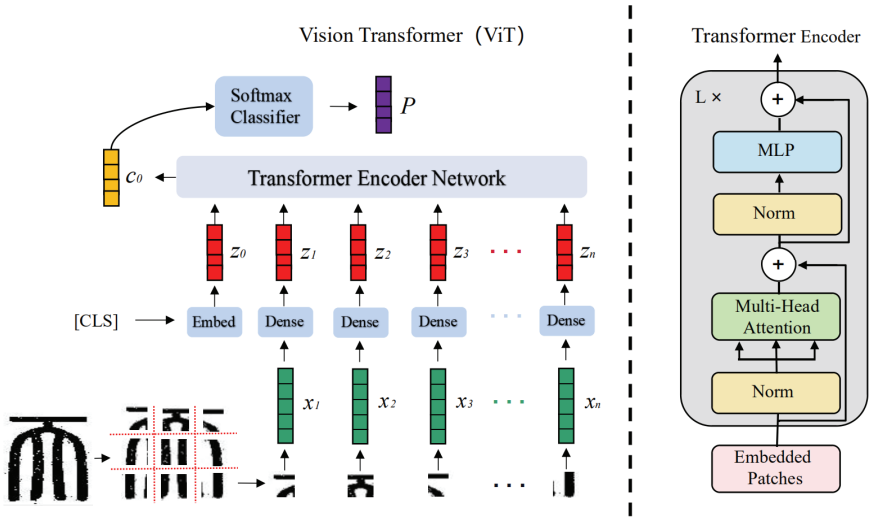


Fig. 5. Vision transformer framework.

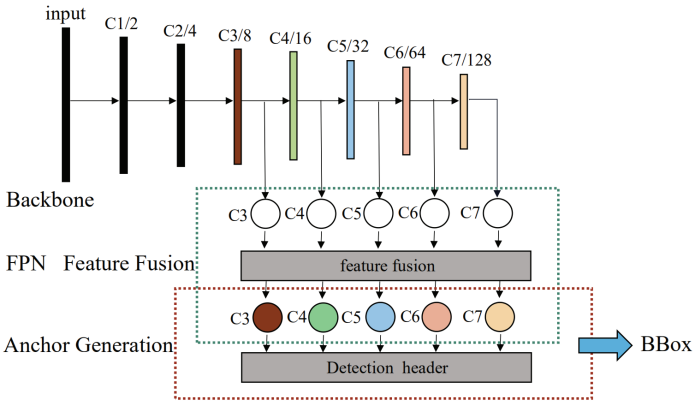


Fig. 6. Framework of FPN.

3.2.4 Improved method SEL-FPN

The number of characters in a seal ranges from 1 to 12 or more. There are more than a dozen seals. The seal image was resized to a fixed size and entered into the backbone network, whereas the sizes of the characters in the same seal are similar. Therefore, more characters means that the character distribution in the seal is denser. The FPN has a superior detection performance for different scaled instances in the same image. This solves the multiscale problem of object detection. In the feature extraction network, the semantic information of low-level features is less, but the target location is more accurate. The semantic information of high-level features is richer, but the target location is rougher. As shown in Fig. 6, the FPN is used to generate candidate object bounding boxes using the RPN. The RPN uses a single-scale convolutional feature map. The FPN is embedded in the RPN network such that feature maps of different scales are generated and fused. Each layer of the FPN defines anchor frames of different sizes, for example, for layers P2, P3, P4, P5, and P6, the size of the anchor frame is $32^2, 64^2, 128^2, 256^2, 512^2$. Each layer had three ratios:1:2, 1:1, and 2:1. Thus, the entire feature pyramid contained 15 types of anchor frames.

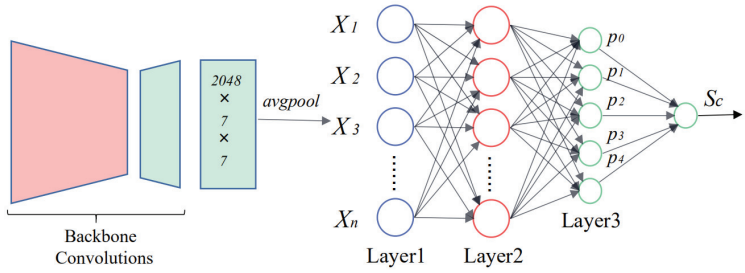


Fig. 7. Framework of SEL-FPN.

To adaptively select the appropriate feature layer for the FPN, an SEL-FPN was proposed in this study. The SEL-FPN selects a specific feature map and uses it as an input for the RPN. Owing to the symmetrical aesthetic characteristics of ancient artwork, the characters in the seal are evenly arranged and fill the entirety of the seal in most cases. Moreover, the scale of the characters in the same seal was essentially the same. The SEL-FPN selects the corresponding feature layer according to the number of characters in the seal. Seals are divided into five classes according to their density such as A: B {1,2: P6; 3,4: P5; 5,6: P4; 7–9: P3; 10 or more: P2}. A represents the number of characters in the seal and B represents the feature layer in the FPN. If there are only one or two characters in the seal, the character density is low, corresponding to the feature map P6 in the FPN. Instead, if there are 10 or more characters in the seal, the character density is high, corresponding to feature map P2 in the FPN. We recorded the feature map in the FPN corresponding to the seals with different character densities as the training set parameters. The SEL-FPN plays a role in scale selection. As shown in Fig. 7, we obtained the output (a 7×7 feature map with 2,048 channels) from the C5 layer of the backbone network ResNet-50. After avgpool, the size of the output becomes $2048 \times 1 \times 1$ which is the input of the subsequent fully connected network. Finally, the output of the fully connected layer is classified by softmax to select the most suitable feature map in the FPN as the input of the RPN. The softmax function is defined as follows:

$$S_C = \frac{e^c}{\sum_{d=1}^{C} e^d} \quad for\ c = 1 \ldots \tag{6}$$

where C is the number of categories, and c is the output value of the ith node. The loss function of the SEL-FPN is:

$$\mathcal{L}_{SEL} = -\sum_{i=1}^{K} y_i \log(p_i) \tag{7}$$

where K denotes the number of classes. y_i is the label, which is equal to 1 if the category is i and 0 otherwise. p_i is the output of the neural network, that is, the probability that the category is i. This output value is calculated using softmax, as shown in Eq. (6).

Therefore, the final loss function of the SEL-RefineMask is defined as:

$$\mathcal{L} = w_1\mathcal{L}_{SEL} + w_2\mathcal{L}_{cls} + w_3\mathcal{L}_{box} + w_4\mathcal{L}^k \tag{8}$$

where w_1, w_2, w_3, w_4 are adjustable weights. If the number of characters in a seal is small, the results of the SEL-FPN were biased toward the higher level, and the anchor frame corresponding to this level is

larger. Otherwise, if the number of characters in a seal is large, the SEL-FPN results were biased towards the lower layer, and the anchor frame corresponding to this layer is smaller.

4. Experimental Results

4.1 Experiment Analysis

We recorded the standard COCO format dataset evaluation criteria, such as AP (averaged over IoU thresholds), AP_{50} and AP_{75} (AP at different scales), and AP on instances of small, medium, and large sizes (labelled as AP_s, AP_m, and AP_l, respectively). In this study, we compared SEL-RefineMask with other instance segmentation algorithms. The results indicated that ResNet-50 is the most suitable backbone network for extracting the seal features. Experiments proved that the proposed SEL-FPN effectively improves the AP of the network and enhances the segmentation performance.

4.2 Comparison with Other Methods

We used Mask R-CNN, PointRend [30], HTC [31], and RefineMask as the baseline methods. A quantitative comparison of SEL-RefineMask with the baseline methods was performed on the SACB dataset. As presented in Table 1, the algorithm used in this study outperformed the other four algorithms when the backbone network was ResNet-50-FPN. Among them, AP, AP_{50}, and AP_{75} improved by 3.19%, 3.03%, and 0.76% over RefineMask, respectively. As the SEL-FPN reduces the number of anchor frames, the runtime of SEL-RefineMask (13.8 fps) was enhanced and better than that of RefineMask (10.2 fps). When the SEL-FPN module was applied, the performance and runtime of Mask R-CNN, PointRend, and HTC were also enhanced.

Table 1. Instance segmentation results

Algorithm	Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	Runtime (fps)
Mask R-CNN	R-50-FPN	45.90	57.66	48.12	19.56	40.23	45.13	14.6
+SEL-FPN	R-50-FPN	47.04	60.28	50.35	21.07	38.16	49.24	16.3
PointRend	R-50-FPN	48.89	57.64	50.22	25.20	38.24	44.11	10.7
+SEL-FPN	R-50-FPN	51.50	60.99	51.95	23.98	40.23	50.11	13.1
HTC	R-50-FPN	50.11	59.30	51.94	23.35	43.44	54.33	4.0
+SEL-FPN	R-50-FPN	53.15	62.64	52.96	19.73	44.56	53.45	6.4
RefineMask	R-50-FPN	51.95	64.93	58.18	25.34	45.29	51.07	10.2
SEL-RefineMask	R-50-FPN	**55.14**	**67.96**	**58.94**	**26.15**	**47.14**	**55.58**	**13.8**

The best results are highlighted in bold.

Table 2. Comparison of different backbone networks

Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
VGG16-FPN	55.86	64.00	55.96	22.44	42.34	54.88
ResNet-34-FPN	52.96	63.51	54.58	25.10	44.56	53.70
ResNet-50-FPN	**55.14**	**67.96**	**58.94**	**26.15**	**47.14**	**55.58**
ResNet-101-FPN	54.17	67.15	56.95	25.49	45.80	54.75

The best results are highlighted in bold.

Table 2 indicates that ResNet-50-FPN performed best among the different feature extraction backbone networks. For the seal dataset with special distribution rules (uniformity and symmetry), ResNet-50-FPN is more suitable for feature extraction than the other backbone networks.

We divided the test set into five parts according to the number of characters and tested them separately. We controlled the output of the SEL-FPN such that the test set can be tested on each feature layer. The results are shown in Fig. 8. For example, a test set containing only one or two characters exhibited the best performance in the P6 layer. Therefore, the anchor produced by the P6 layer has a higher quality and usability than the other layers. The test results on the other four datasets with the same number of characters also confirmed this. Consequently, the SEL-FPN effectively improved the segmentation accuracy by selecting the feature layer in the FPN according to the number of characters in the seal. The accuracy rate of SEL-FPN in predicting the number of characters in the seal recorded during the experiment reached 94.28%.

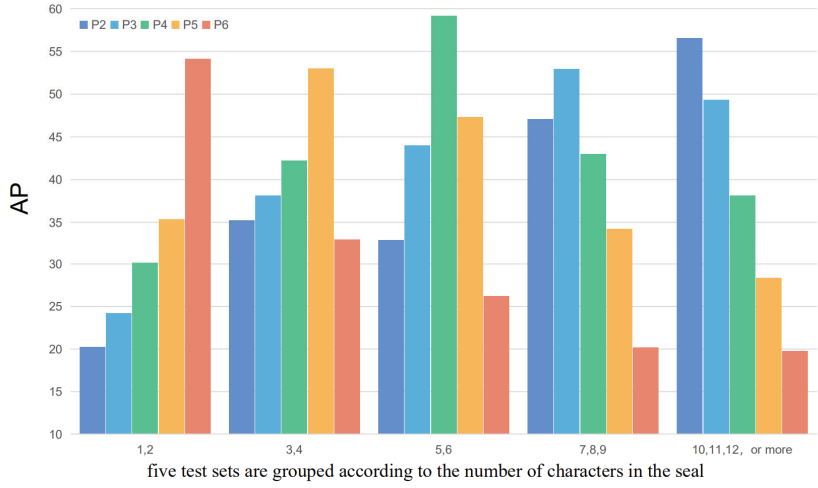


Fig. 8. Test results on different layers of FPN for the test set divided by the number of characters.

The existing mask-generating branch of SEL-RefineMask predicts foreground and background labels when the box branch generates a box for each class (via a pixel-level sigmoid binary loss function). This does not require competition between the classes. In Table 3, it was compared with a pixel-level polynomial loss softmax function (e.g., FCN), which requires competition between classes. The method using the pixel-level polynomial loss softmax function completed mask and class prediction tasks together, which resulted in a severe loss of 6.1%, 7.78%, and 6.9% for the AP, AP_{50}, and AP_{75}, respectively. This indicates that in the training task, once the character instances are grouped as a whole (obtained from the box branch prediction), it is sufficient to predict a binary mask without considering the classes, and thus easier to train.

We also compared RoIPool, RoIWarp, and RoIAlign, and the experimental results are presented in Table 4. In the character segmentation experiment, ResNet-50 was used as the backbone network. RoIAlign's RoI mapping from the original image to the feature map uses bilinear interpolation directly without rounding. The loss of pixel information is reduced during the instance segmentation process of the seal and the accuracy of the corresponding return to the original image after pooling is improved. Compared to RoIPool, AP of RoIAlign improved by approximately 3%, and the improvement of

RoIAlign with average pooling relative to maximum pooling is limited. Both RoIWarp and RoIAlign use bilinear sampling, but RoIWarp does not align RoI, which also supports that the alignment strategy of RoIAlign helps enhance the accuracy of the segmentation.

Table 3. Comparison of softmax and sigmoid loss functions

	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Softmax	49.04	60.18	52.04	25.12	44.83	52.16
Sigmoid	55.14	67.96	58.94	26.15	47.14	55.58
Δsoftmax–sigmoid	+6.10	+7.78	+6.90	+1.03	+2.31	+3.42

Table 4. Comparison of RoIPool, RoIWarp, and RoIAlign operations

	Align	Bilinear	agg.	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
RoIPool			max	52.03	64.33	50.82	17.24	32.78	40.25
RoIWarp		✓	max	53.15	65.01	52.16	19.16	35.10	39.80
		✓	ave	51.49	64.33	50.99	18.25	42.88	48.14
RoIAlign	✓	✓	max	55.04	67.90	58.71	24.13	45.34	50.95
	✓	✓	ave	**55.14**	**67.96**	**58.94**	**26.15**	**47.14**	**55.58**

The best results are highlighted in bold.

We compared the proposed method with traditional digital image processing methods, such as the drip, improved drip, and projection methods. We selected 3,000 seal images as the test set. The smallest box containing characters was marked as M, the prediction box was marked as N, and the threshold η was set to 0.9. When $(M \cap N)/M > \eta$ && $(M \cap N)/N > \eta$, the prediction box is suitable, and the comparison results are listed in Table 5. Based on the results, the SEL-RefineMask is superior to traditional methods.

Table 5. Comparison of traditional methods and SEL-RefineMask

Methods	Accuracy (%)	Number of sticky characters
Number of images	3,000	3,000
SEL-RefineMask	84.03	89
Drip method	48.89	215
Improved drip method	75.44	170
Projection method	78.34	135

We also selected 3,000 characters as the test set to analyze the ability of the proposed and traditional methods to segment sticky characters. Table 5 lists the number of sticky characters using different methods. The proposed method has the least number of sticky characters.

For the hyperparameters, the initial learning rate was set to 0.02. The proposed model was trained on 8 GPUs, and the mini-batch size was set to 2. The proposed model trained for 10k epochs, and the learning rate was set to 0.002 when iterating to 8k epochs. The weight decay was set to 0.0001 and the momentum was set to 0.9. w_1, w_2, w_3 and w_4 of the loss function were set to 1:5:50:40. This also indicates that the last two items of the loss function account for a small proportion and require larger weight parameters for adjustment.

After segmentation, thousands of classifications of the SSF were performed. In this study, SVM, Alex Net, VGG16, ResNet-50, ResNet-101, ResNeSt-50, ResNeSt-101, and ViT were trained and tested on the SSF dataset (containing 1,000 categories and 10,000 small seal font images), and the results are

illustrated in Fig. 9. According to the results analysis, the classification of convolutional network for images were superior to those of SVM, and the ViT classification accuracy reached 91.0%. This proved that for small seal characters with complex strokes, the attention mechanism in the network improves the classification accuracy. Another important reason is that the ViT is pretrained on a large dataset.

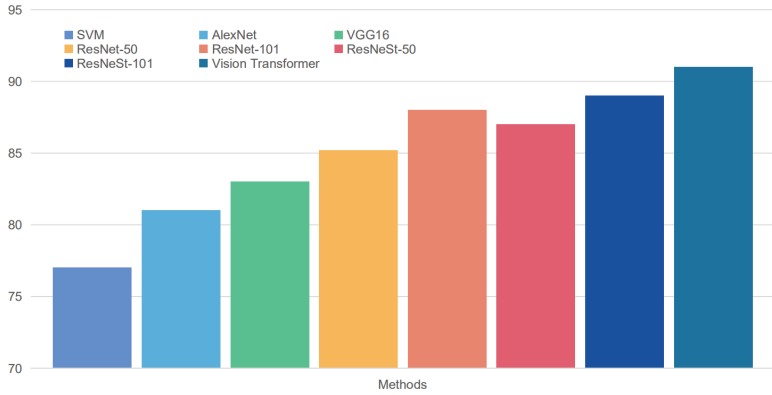


Fig. 9. Character recognition results.

To verify the effectiveness of the proposed method using SEL-RefineMask and evaluate the qualitative analysis performance of the method, we selected unlabelled seal images for testing. The results of comparing the SEL-RefineMask algorithm with the other algorithms are presented in Fig. 10. The mask generated by the mask branch is shown in the figure, whereas the regression box is not.

The original seal and the segmentation results of the SEL-RefineMask, RefineMask, and Mask R-CNN are illustrated in Fig. 10. SEL-RefineMask can segment the outline of characters clearly, and Mask R-CNN recognizes the boundary of the seal as part of the character, as shown in the right corner of Fig. 10. RefineMask and Mask R-CNN always treat two or more characters as one because SEL-FPN is not used. Through analysis, the SEL-FPN can select the most suitable feature layer for seal segmentation to generate anchors through training. In comparison to other algorithms, the proposed algorithm achieved a superior performance of de-bordering while accurately segmenting characters, which can generate a higher quality segmentation mask for each small seal font character instance and achieves a good segmentation result at a semantic level. Through experimental analysis, seal segmentation and character recognition results were found to be more accurate. The character instances (obtained by segmentation of SEL-RefineMask) were classified using the ViT classification network to obtain simplified Chinese characters corresponding to the small seal script.

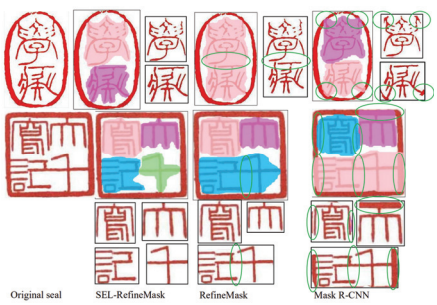


Fig. 10. Test results samples.

5. Conclusion

This study presented a method for segmenting and recognising characters in ancient Chinese seals. The segmentation module was an improvement of RefineMask, and the character recognition module used was ViT. Moreover, this study created two publicly available datasets for seal character segmentation and recognition. We compared the proposed method with other baseline methods to verify its reliability. The experimental results indicated that the proposed method outperforms other generally used methods. However, the proposed solution still has the following shortcomings: (1) In the experiment, the segmentation of sticky characters and the removal of borders were difficulties in the research. Although the proposed method outperformed the other methods in these two aspects, these two problems still exist. (2) The generalisation ability of the model needs improvement. To address these limitations in future research, novel solutions will be studied, expansion of training set will be explored, and continuous improvement of the proposed method will be implemented such that it can be applied to other text recognition scenarios. The recognition of features such as carving strokes, grammar, and the carving style of ancient characters in seals will be the main focus of our next work. The super pixel information can be used to detect and classify objects in an image based on their location [32]. In the future, we can identify and classify cultures and genres based on the characteristics of seal-engraving. Eventually, we plan to construct historical transmission information of ancient texts based on seals and provide data support and a research basis for historical and cultural exchanges.

Acknowledgement

This study was supported by Shandong Provincial Natural Science Foundation (ZR2020MA064). We thank the original seal data provided by digital platforms such as Jangseogak of the Academy of Korean Studies and the Kyujanggak Institute for Korean Studies.

References

[1] S. Liang, “Analysis of the role of seal identification on calligraphy and painting identification,” *Identification and Appreciation to Cultural Relics*, vol. 2021, no. 23, pp. 96-98, 2021.

[2] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, “Detecting text in natural image with connectionist text proposal network,” in *Computer Vision – ECCV 2016*. Cham, Switzerland: Springer, 2016, pp. 56-72.

[3] S. Tangwannawit and W. Saetang, “Recognition of lottery digits using OCR technology,” in *Proceedings of 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Naples, Italy, 2016, pp. 632-636.

[4] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, “PhotoOCR: reading text in uncontrolled conditions,” in *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia, 2013, pp. 785-792.

[5] B. Su and S. Lu, “Accurate scene text recognition based on recurrent neural network,” in *Computer Vision – ACCV 2014*. Cham, Switzerland: Springer, 2014, pp. 35-48

[6] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[7] T. Jiang, M. Qiu, J. Chen, and X. Cao, "LILA: a connected components labeling algorithm in grid-based clustering," in *Proceedings of 2009 1st International Workshop on Database Technology and Applications*, Wuhan, China, 2009, pp. 213-216.

[8] L. Xu, F. Yin, Q. F. Wang, and C. L. Liu, "Touching character separation in Chinese handwriting using visibility-based foreground analysis," in *Proceedings of 2011 International Conference on Document Analysis and Recognition*, Beijing, China, 2011, pp. 859-863.

[9] R. Ma and J. Yang, "An improved drop-fall algorithm for handwritten numerals segmentation," *Journal of Chinese Computer Systems*, vol. 28, no. 11, pp. 2110-2112, 2007.

[10] Q. Hu, J. Yang, Q. Zhang, K. Liu, and X. Shen, "An automatic seal imprint verification approach," *Pattern Recognition*, vol. 28, no. 8, pp. 1251-1266, 1995.

[11] K. Li, B. Batjargal, and A. Maeda, "Character segmentation in Asian collector's seal imprints: an attempt to retrieval based on ancient character typeface," *Journal of Data Mining and Digital Humanities*, 2021. https://doi.org/10.46298/jdmdh.6102

[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1-20, 2016.

[13] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: an end-to-end trainable neural network for spotting text with arbitrary shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 532-548, 2021.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 2980-2988.

[15] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting with convolutional recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 5238-5246.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014 [Online]. Available: https://arxiv.org/abs/1409.0473.

[17] A. Lamb, T. Clanuwat, and A. Kitamoto, "KuroNet: regularized residual U-Nets for end-to-end Kuzushiji character recognition," *SN Computer Science*, vol. 1, no. 3, pp. 1-15, 2020.

[18] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: fast oriented text spotting with a unified network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 5676-5685.

[19] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, et al., "ICDAR 2015 competition on robust reading," in *Proceedings of 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015, pp. 1156-1160.

[20] ICDAR 2017 Robust Reading Competitions [Online]. Available: http://rrc.cvc.uab.es/.

[21] M. Aamir, Z. Rahman, W. A. Abro, M. Tahir, and S. M. Ahmed, "An optimized architecture of image classification using convolutional neural network," *International Journal of Image, Graphics and Signal Processing*, vol. 10, no. 10, pp. 30-39, 2019.

[22] K. Wada, "Labelme: image polygonal annotation with Python," 2016 [Online]. Available: https://github.com/wkentaro/labelme.

[23] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar and C. L. Zitnick, "Microsoft coco: common objects in context," in *Computer Vision – ECCV 2014*. Cham, Switzerland: Springer, 2014, pp. 740-755.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 770-778.

[25] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 936-944.

[26] G. Zhang, X. Lu, J. Tan, J. Li, Z. Zhang, Q. Li, and X. Hu, "RefineMask: towards high-quality instance segmentation with fine-grained features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual Event, 2021, pp. 6861-6869.

[27] 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," 2021 [Online]. Available: https://arxiv.org/abs/2010.11929.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91-99, 2015

[29] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1440-1448.

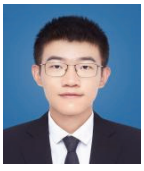
[30] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, et al., "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, pp. 4974-4983.

[31] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: image segmentation as rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 2020, pp. 9796-9805.

[32] M. Aamir, Y. F. Pu, Z. Rahman, W. A. Abro, H. Naeem, F. Ullah, and A. M. Badr, "A hybrid proposed framework for object detection and classification," *Journal of Information Processing Systems*, vol. 14, no. 5, pp. 1176-1194, 2018.

Ze-dong Dun **https://orcid.org/0000-0001-6345-6660**

He received his B.S. degree from the National Pilot School of Software, Yunnan University in 2019. He is currently pursuing his M.S. degree at the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai. His current research interests include image processing, computer vision, and artificial intelligence.



Jian-yu Chen **https://orcid.org/0000-0002-0060-3843**

He is pursuing his B.S. degree at the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai. His current research interests include astronomical image processing and machine learning algorithms.

Mei-xia Qu **https://orcid.org/0000-0001-7607-8195**

She received her Ph.D. from the School of Computer Science and Technology, Shandong University. She is currently a lecturer at the Department of Computer Science at Shandong University, Weihai. Her research interests include machine learning, artificial intelligence, and big data research.

Bin Jiang **https://orcid.org/0000-0002-2897-5745**

He received his Ph.D. from the University of Chinese Academy of Sciences. He started working since May 2005 and now he is an associate professor at the Department of Computer Science at Shandong University, Weihai. His current research interests include astronomical data mining and machine learning.