Hindawi
Security and Communication Networks
Volume 2022, Article ID 8032726, 10 pages
https://doi.org/10.1155/2022/8032726

Research Article

Invoice Detection and Recognition System Based on Deep Learning

Xunfeng Yao, Hao Sun, Sijun Li, and Weichao Lu

Jinling College, Nanjing University, Nanjing, China

Correspondence should be addressed to Xunfeng Yao; 030504@jlxy.nju.edu.cn

Received 13 August 2021; Accepted 29 September 2021; Published 25 January 2022

Academic Editor: Xuyun Zhang

With the development of economy and information technology, a large amount of invoice information has been produced. As one of the important components of the industrial Internet of Things, the recognition of invoice information is urgent to realize its intelligent recognition. Most invoice issuing units basically adopt traditional manual identification methods for the processing of invoices. As the number of invoices increases, problems such as low efficiency in identifying invoice information, error-prone, and difficulty in ensuring security frequently appear. In response to the above problems, this paper designs and implements an invoice information recognition system based on deep learning. The system first solves the problems of low image contrast and lack of image due to poor lighting or noise effects by image preprocessing methods such as image graying and normalization. Second, a target detection and invoice recognition method based on the combination of YOLOv3 + CRNN two models is proposed, and an end-to-end invoice information recognition model is obtained. Finally, the model is used to develop an invoice detection and recognition system based on deep learning. Experiments have verified that the system has the characteristics of high recognition accuracy and high efficiency, which can accurately identify invoice content information and reduce the loss of manpower and material resources.

1. Introduction

Foreign research on invoice recognition system originated in the 1960s and 1970s. But, most research is only on methods of invoice recognition and digital recognition. The concept of OCR (Optical Character Recognition) technology was first proposed by German scientist Tausheck in 1929. As an important part of pattern recognition, OCR is used to identify the information in the image and extract it into computer readable [1]. Until about the 1960s, Japan began to study the basic recognition theory of OCR. After more than ten years of research, it developed a simple recognition system such as postal code recognition, which realized the automatic recognition of codes on mails [2]. After 1970, China began to study OCR technology and first carried out relevant research on Chinese character recognition. Until 1986, Tsinghua University and other universities developed an invoice recognition system based on OCR technology, and Chinese OCR invoice recognition products came out [3]. Due to the low recognition rate of the early invoice system and insufficient productization, it has not been popularized in life. With the rise of artificial intelligence, more systems for invoice recognition have begun to appear on the market. For example, Baidu's OCR recognition system and Tencent's OCR recognition system both use in-depth learning to detect and recognize invoice information [4].

Once deep learning has emerged, it has been widely used in speech recognition, image recognition, and natural language processing. In 2011, Google applied deep learning to speech recognition and successfully reduced the error rate [5, 6]. In the field of image recognition, researchers have further proposed a large-scale deep convolutional neural network, which reduces the error detection rate to 15.3% [7]. In 2015, He et al. proposed the ResNet architecture to improve the accuracy of the algorithm by increasing the amount of data during training [8]. Deep learning has developed rapidly in image recognition, and target detection technology has been applied to text localization in natural scenes. Girshick et al. proposed that R-CNN successfully

applied deep learning to target detection. First, the selective search algorithm was used to select candidate boxes, and then the candidate boxes were sent to the convolutional neural network for classification, but the extracted candidate boxes overlapped a lot, and feature extraction redundancy exists [9, 10]. Later, the improved FastR-CNN algorithm in the research inputs the entire image into the convolutional neural network and then maps the candidate frame on the feature map, avoiding repeated feature extraction and improving the training speed [11]. The concept of anchor frame is proposed in Faster R-CNN in [12], and the extraction of candidate frames is also realized by convolutional network, which effectively reduces the selection time of candidate frames. Dai et al. proposed to integrate the target location information into the ROI pooling layer to construct a location-sensitive score map, which effectively solves the problem of the destruction of the translation invariance of the convolutional network [13]. In order to adapt features to targets of different sizes, Lin et al. proposed a feature pyramid structure for small target detection [14]. Although the above-mentioned algorithm has high detection accuracy, it cannot achieve a real-time effect. In order to solve the efficiency problem, Redmon et al. proposed to use a single-structure convolutional neural network to directly predict the location and category of the target, but the accuracy is slightly lower [15]. Later, an improved YOLOv2 algorithm [16] was proposed, and a batch normalization layer [17] (Batch Normalization) was added on the basis of YOLOv1 to speed up training, and anchor boxes and higher resolution classifiers were used to improve accuracy. Literature [18] improved the YOLOv2 network by changing the screening rules of the candidate frame and other methods and achieved relatively ideal results in the task of positioning the invoice recognition image. In 2014, Liu et al. proposed the SSD algorithm, which takes into account both speed and accuracy, but the shallow feature expression ability of its prediction layer is not strong [19]. In order to strengthen the expressive ability of shallow features, Fu et al. proposed to use deconvolution to add contextual information to the feature map, and the accuracy of the model was further improved [20]. After YOLOv2, Redmon proposed the third version of the YOLO series, YOLOv3 [21]. This algorithm uses Faster R-CNN to extract features to improve the speed of target detection, which is very suitable for natural scenes with multiple anticounterfeit feature detection in invoices.

In order to meet the requirements of efficiently identifying invoice data in engineering applications, this paper first uses the YOLOv3 algorithm for text target detection training. Second, the deep learning CRNN model is used to identify the content of the invoice. Finally, the two models are combined to obtain an end-to-end invoice recognition model, which is verified by the test set, and the recognition result is compared with the recognition result of the traditional OCR technology.

2. Invoice Recognition System Based on Deep Learning

2.1. Invoice Detection Based on YOLOv3 Algorithm. The YOLO algorithm (You Only Look Once, YOLO) is a neural network model that can identify and detect objects and text. The execution process of the algorithm is mainly divided into two parts: (1) first classify the object; (2) identify the position of the object in the picture. Because YOLO's unique end-to-end design method simplifies object detection into a single regression problem, it avoids the problem of slow running speed and difficult model convergence. The emergence of the YOLO algorithm gives new ideas to the target detection task. The algorithm combines the two tasks of positioning and classification to make the image detection speed meet the requirements of real-time detection. YOLO is composed of four parts: input layer, convolution layer, pooling layer, and fully connected layer. Its network structure model is shown in Figure 1.

YOLO extracts the feature value through the convolutional layer CNN, and the final result of the predicted value is completed through the fully connected layer. As shown in Figure 1, among the 24 convolutional layers, channel reduction is first performed by 1×1 convolution, and then 3×3 convolution processing is used. In the convolution and fully connected layers, Leaky ReLU is used to activate the function: max (x, 0.1x), and YOLO uses a mean square error loss function. The positioning error refers to the error of the bounding box coordinate prediction. The calculation of the error uses a weight value of $\lambda_{\text{co ord}} = 5$. The confidence of the bounding box containing the target and the bounding box not including the target is calculated, and the other weight values are set as 1, using the mean square error of the model as the loss function. For bounding boxes with inconsistent sizes, the actual smaller bounding box is more sensitive. In order to solve the above phenomenon, the model changes the predicted value to $(x, y, \sqrt{w}, \sqrt{h})$. The principle is that each cell can predict multiple bounding boxes, and each bounding box corresponds to the corresponding category. Select the bounding box with the largest IOU with the ground truth for the task of predicting the target. Other bounding boxes will ignore the existence of the target to obtain a specialized cell corresponding to the bounding box. For the bounding box that does not have a corresponding target, its error term is confidence. The loss function formula of YOLO is shown in the following formula:

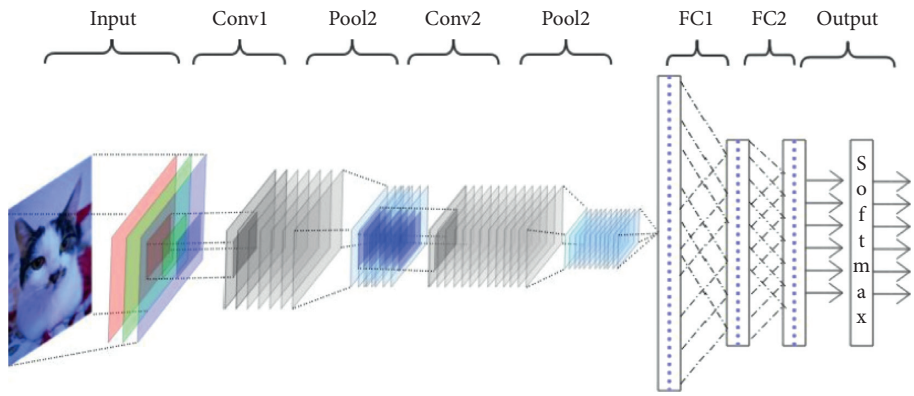

Figure 1: YOLO network structure diagram.

$$\begin{aligned}\text{loss} = {} & \lambda_{\text{co ord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{co ord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 + \lambda_{\text{co ord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 + \sum_{i=0}^{S^2} 1_{ij}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2.\end{aligned} \tag{1}$$

The YOLOv3 algorithm used in this paper combines the advantages of other algorithms with the YOLO algorithm so that the algorithm further improves the accuracy of model detection while maintaining the speed advantage, especially for small target tasks. The improvement of the detection ability is more obvious. YOLOv3 improves model performance mainly by adjusting the network structure. This method uses the new backbone network Darknet-53, and at the same time, it uses multiscale features to detect target tasks by constructing an FPN network. The Darknet-53 network structure has 53 convolutional layers, which combines the advantages of the residual network with shortcut connections between some layers. The specific structure of Darknet-53 is shown in Figure 2.

YOLOv3 mainly uses the remaining 52-layer network structure of darknet-53 except for the fully connected layer. In order to improve the accuracy of the algorithm for detecting small target tasks, YOLOv3 uses a fusion method similar to FPN to perform detection on multiple scale feature maps. The 3 prediction routes of YOLOv3 are for three convolutional structural layers; the number of convolution kernels in the last convolutional layer is 255, which is for the 80 categories of the COCO data set: $3 * (80 + 4+1) = 255$, where 3 represents that a grid cell contains 3 bounding boxes, 4 represents the 4 coordinate information selected by the box, and 1 represents the objectness score. In the Darknet-53 network, $256 * 256 * 3$ is used as input, and the leftmost column of numbers represents repeated residual components. Each residual component has two convolutional layers and a shortcut link. The residual component of the specific direct connection method is shown in Figure 3. Input x to the output process, and the output result is $f(x) + x$. When $f(x) = 0$, $H(x) = x$, at this time, the residual result approaches 0, and the model converges.

YOLOv2 uses the pass-through structure to identify and detect fine-grained features, while the YOLOv3 method uses three different scale feature maps to identify and detect objects based on the YOLOv2 method. Among them, in the first scale, some convolutional layers are added after the traditional basic network for sampling, the sampling multiple is high, and the perception field is large, so this scale is suitable for large object detection; the second scale is from the 79th layer upwards; convolution and sampling are added to the last 16×16 feature map. This scale is suitable for medium-sized object detection; the third-scale tree uses a 32×32 feature map, which is suitable for small object detection. YOLOv3 extends the K-means clustering of YOLOv2, which takes the form of a priori frame size, sets 3 a priori frames for each downsampling scale, and finally clusters a priori frames of 9 sizes. See Table 1 for the specific allocation of a priori boxes of 9 scales.

As shown in Table 1, when the feature map is on a 13×13 feature map with a larger receptive field, a larger prior frame is needed to detect a larger target. When on a 26×26 feature map, a medium a priori box needs to be used to detect medium-sized objects. When on the 52×52 minimum receptive field feature map, a smaller prior frame is needed to detect smaller objects.

Unlike YOLOv1 and YOLOv2, which both use the mean square error as the loss function, YOLOv3 uses the cross-entropy loss function to calculate the coordinate loss. YOLOv3 improves the category prediction function, and the softmax layer will no longer be used. The essence of the softmax layer in the classification network is that a category contains an attribute, such as an image or an object. But when in a complex scene, an object can contain multiple categories. For example, there are two categories of woman and person in the category of people, and there is a woman in an image, which corresponds to the category label in the

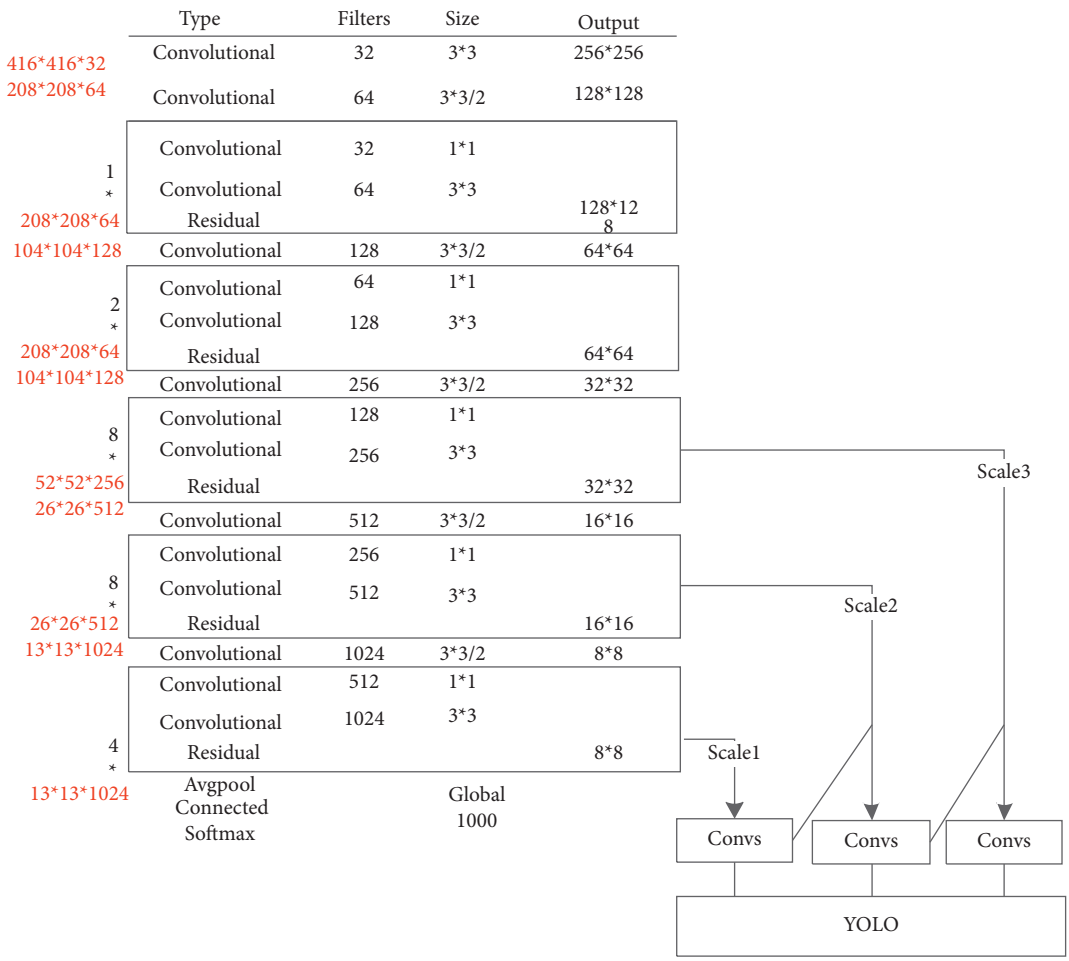

Figure 2: Darknet-53 structure details.

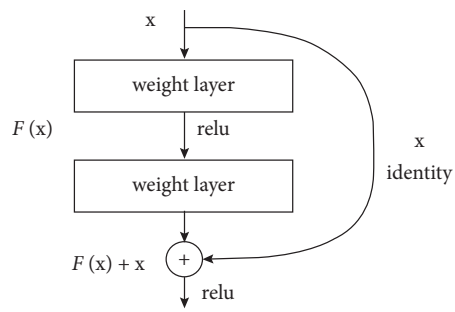

Figure 3: Schematic diagram of the residual group structure.

Table 1: A priori box allocation resources.

Feature map	13 * 13			26 * 26			52 * 52		
Receptive field	Big			Middle			Small		
Anchor	116 * 90	156 * 198	373 * 326	30 * 61	62 * 45	59 * 119	10 * 13	16 * 30	33 * 23

model detection result. There are two classes of woman and person at the same time, which belong to the multilabel classification. For this type of problem, softmax will choose the category with the largest prediction probability, which will eventually result in only one category being detected by woman and person. In order to solve the above problems, YOLOv3 uses a logistic regression layer for classification to obtain different categories. The logistic regression layer uses the sigmoid function. The sigmoid function can control the output between 0 and 1. Therefore, the sigmoid function is

used to control the output of a certain category after the feature is extracted. The value is greater than 0.5. It can be seen that this category belongs to this category; otherwise, it does not belong to this category, so that a box can predict multiple categories in this image, and the cost function here is the cross entropy of sigmoid. The IoU loss function and focal loss are used in the YOLOv3 target detection algorithm, and 1-GIoU is directly used as the bounding box regression loss function to replace the original mean square error and loss function. The focal loss based on the cross-entropy loss is used as the loss function of the confidence of the bounding box object. The target classification loss uses the classical cross entropy as the loss function, using the GIoU loss function and the focal loss function, and the resulting YOLOv3 loss function is shown in the following formula:

$$\begin{aligned}\text{loss} &= b\,\text{boxloss} + \text{confidenceloss} + \text{classloss}\\ &= \sum_{0}^{\text{cell\_number\_}B} I^{\text{object}} \times \left(1 - GIoU_{\text{predict}}^{\text{groud\_truth}}\right)\\ &\quad + \sum_{0}^{\text{cell\_number\_}B} m \times \text{focal\_loss}\left(CE\left(p_0, q_0\right)\right)\\ &\quad + \sum_{0}^{\text{cell\_number\_}B} I^{\text{object}} \times \sum_{0}^{c} CE\left(p(c), q(c)\right).\end{aligned} \tag{2}$$

In the bounding box regression loss function (bboxloss) part, the original mean square error and loss function are replaced by the GIoU loss function. The loss function also adds the focus loss to the boundary box confidence cross entropy loss function, so as to balance the loss proportion of easy samples and difficult samples.

2.2. CRNN-Based Invoice Edge Detection and Recognition. In order to accurately extract the information in the invoice, it is necessary to detect and identify the boundary of the detected invoice. What edge detection can do is to identify the points with obvious brightness changes in the digital image. This process can discard the redundant information in the image, thereby reducing some unnecessary processing. This method improves the information extraction rate. At the same time, important boundary information in the image is retained. This paper uses the CRNN (Convolutional Recurrent Neural Network) to detect the edges of invoices. This network recognizes text sequences of variable length end-to-end, so there is no need to cut individual texts in advance but convert text recognition into a sequence learning problem dependent on timing, that is, image-based sequence recognition. According to the characteristics of Chinese handwriting recognition, this paper improves the CRNN network in order to solve the problem of Chinese handwriting recognition. The network structure of the improved CRNN is composed of three parts, which from bottom to top are the Deep Convolutional Layer, Recurrent Layer, and Transcription Layer. The structure of CRNN is shown in Figure 4.

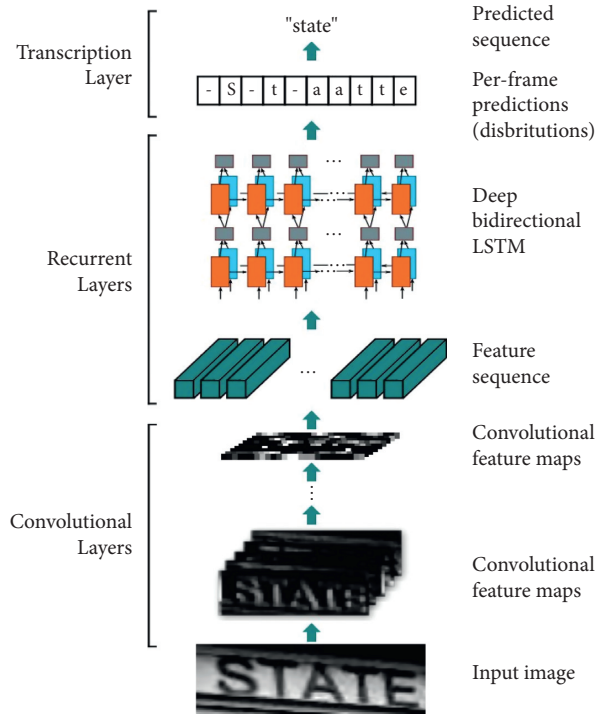

Figure 4: Schematic diagram of CRNN structure.

The specific steps of CRNN model training are as follows:

Step 1. According to the input requirements of the CRNN 7 model, process the data, generate a large batch of invoice sample data, divide the data into the test set and the training set according to the ratio of 9 : 1, and label the image according to the requirements of the CRNN training set.

Step 2. According to the generated data set, set the three major parameters.

Step 3. Design the loss function calculation method in training.

Step 4. Design the entire network training process, loop each epoch, and perform model verification and model storage when the specified number of iterations is reached. The specific process is shown in Figure 5.

3. Experiment

In this paper, the invoice image data is obtained through a scanner device. However, in the process of collecting data, the invoice image will have some noise effects due to environmental changes and improper human operations. Therefore, this paper first preprocesses the invoice data. The process of image preprocessing in this paper includes operations such as normalization, grayscale, and edge detection of the original image, and finally, a binary image whose size meets the model input is obtained through the preprocessing operation. Since the invoice is usually placed randomly by hand during scanning, there will be a certain angle of

Figure 5: Edge detection step process.

inclination. In order to reduce the influence on the subsequent information area positioning and character cutting, this paper uses the Hough line detection method to correct the inclination of the invoice image.

3.1. Experimental Environment. The experiment in this paper is to complete programming and testing on a PC, and the operating system is Windows 7, 64 bits. The programming language is C++, and the system interface is built with MFC. The OpenCV library is needed for image processing, and the LibXL library is needed for data logging to Excel. The experimental environment of this system is shown in Table 2.

OpenCV library is an open-source machine vision development library commonly used at present, which already contains many general algorithms, and the image processing of this system is used for development. MuPDF library is a powerful PDF parser. It is used in this system to convert scanned pdf format images into jpg image format. LibXL library is a package library that implements Excel operations, which is used to automatically save the information recognition results to an Excel table.

3.2. Collection of Data Sets. The collection of invoice images is the initial step of the operation of the entire system. There are generally three methods for collecting invoice images. The first is to capture dynamic video through a camera. This method obtains the invoice data picture by intercepting the invoice information in the video. This process is time-consuming and laborious and the final image obtained is not high-definition; the second one is to collect still images with a high-definition digital camera. This method will generate different edge background information due to different shooting angles or heights; the third is to scan the invoice into a color, grayscale, or binary image through a scanner device. In order to improve the efficiency of invoice recognition and reduce the expenditure of manpower and material resources, this paper chooses a high-definition scanner to obtain invoice mages. The image obtained by the scanner can be saved as a color image, gray image, or binary image. Although the color image is the closest to the real scene, it contains a huge amount of information and a complex color model. It will increase the amount of calculation and time overhead when processing the image, so in general, it is not saved as a color image after scanning. The grayscale image has only one sample color, and the original unclear area in the image can be made clearer through image enhancement technology, and the uninteresting area can also be suppressed. Therefore, the grayscale image is also the input image that people often choose as image processing. All pixels in a binary image have only two values, 0 and 1. Therefore, the data type in the computer generally only occupies 1 binary bit. Comparing the above three images, the color image contains too much information, and the calculation speed is slow; although the calculation of the binary image is simple, the digital information in the invoice image is generally relatively small, and some useful information will be lost after the binarization process. The degree map is a compromise between the two. In order to take into account the recognition rate of numbers and the speed of the system, this paper chooses to save the collected invoice images as grayscale images. This paper uses a D16A3 Jieyu high-speed scanner, which is a high-definition high-speed scanner with a resolution of 4608×3408 dpi and uses the BMP image format for scanning. The pictures collected by this scanner are shown in Figure 6.

3.3. Data Normalization. Since the image of the invoice is obtained by manually operating the scanner when the image is collected, the size of the image obtained by scanning in different environments is different. In order to facilitate the follow-up model to monitor and identify the invoice data information, this paper normalizes the invoice data uniformly.

Image normalization methods mainly include linear and nonlinear processing methods. The advantage of the linear normalization method is that it can retain the linear nature of the original image to a certain extent. The nonlinear normalization will change the quality center of the image and affect the recognition accuracy. Therefore, this paper uses bilinear interpolation to normalize the invoice image to a size of 1245×730.

The bilinear interpolation is shown in Figure 7. The target pixel point $R(i, j)$ is obtained by bilinear interpolation. The four points in the original image are known to be $A11(i1, j1)$, $A11(i1, j2)$, $B21(i2, j1)$, and $B22(i2, j2)$. The principle of using bilinear interpolation to normalize the image is as follows:

$$
\begin{aligned}
f(i, j_1) &= \frac{i_2 - i}{i_2 - i_1} f(i_1, j_1) + \frac{i - i_1}{i_2 - i_1} f(i_2, j_1), \\
f(i, j_2) &= \frac{i_2 - i}{i_2 - i_1} f(i_1, j_2) + \frac{i - i_1}{i_2 - i_1} f(i_2, j_2), \\
f(i, j) &= \frac{j_2 - j}{j_2 - j_1} f(i_1, 1) + \frac{j - j_1}{j_2 - j_1} f(i, j_2).
\end{aligned} \tag{3}
$$

The background in the invoice image is more complicated, and the character spacing is small, which brings greater difficulties to positioning and character cutting. Therefore, this paper designs a fast and accurate positioning and cutting information area positioning algorithm. The specific algorithm flow is shown in Figure 8.

Table 2: Experimental environment table.

Computer configuration	Portable PC, CPU clocked at 2.0 MHz, memory 4G, 64-bit operating system
Operating system	Windows 7
Development environment	VS2013, MFC, and caffe
Open-source library	OpenCV 2.4.10, MuPDF library, and LibXL library

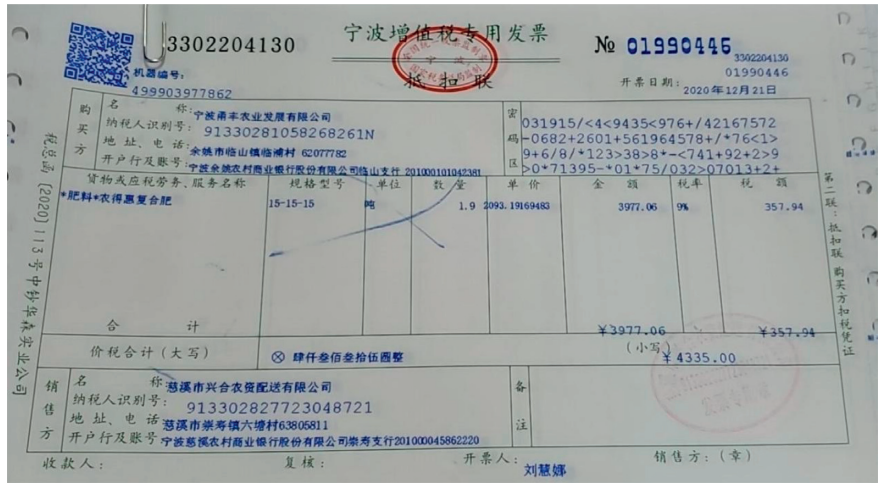

Figure 6: Schematic diagram of invoice data.

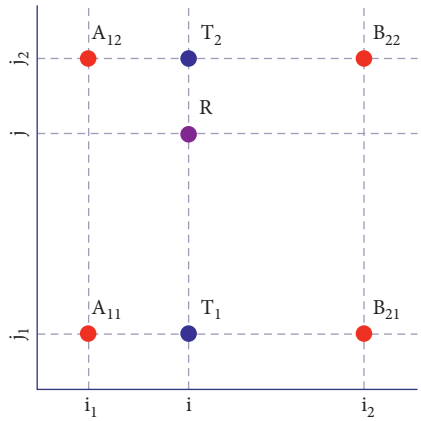

Figure 7: Bilinear interpolation.

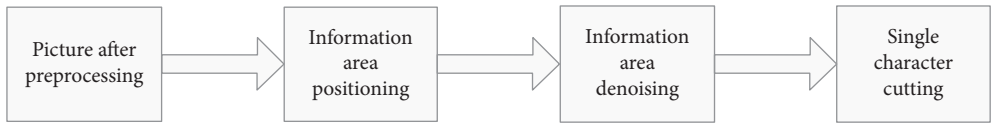

Figure 8: Information area positioning algorithm flow.

Information area positioning is to analyze and locate according to the characteristics of the invoice layout and extract useful information areas such as invoice numbers from the invoice image after data preprocessing. The location of the information area in this paper is mainly based on the characteristics of each functional unit on the invoice page, and a large amount of prior knowledge is obtained through experiments to realize the extraction of the information area. First, use the characteristic of each information area to have a fixed position in the invoice layout, combined with prior knowledge, directly obtain the rough positioning of each information area, then use the symbolic features contained in each information area, and use the method of template matching to compare the roughly extracted information. The area is further accurately positioned, and accurate digital string information is obtained [22].

Information area positioning is a crucial step in the image recognition of invoices. After the previous tilt

Figure 9: Information area positioning.

correction processing, the next step is to extract useful information areas from the entire invoice image to facilitate subsequent character cutting and recognition operations. As value-added tax invoices are unified across the country and have the same and fixed layout structure, the characteristics of the invoice structure can be used to obtain useful information areas. This paper realizes the information extraction of the amount, tax amount, taxpayer identification number, and invoice number, as shown in Figure 9.

The amount and the RMB symbol in the tax information area are matched, the standard template prepared in advance is imported into the database, and the target image uses the matchTemplate() function provided by OpenCV to match the image area that overlaps the template. The result of template matching is shown in Figure 10.

The red rectangular box in Figure 10 is the matching result of the standard squared deviation matching method. By analyzing the digital information of the amount and tax area in the invoice image, it can be seen that the height of the numbers in these two areas is similar to the height of the RMB symbol, and the distance between the RMB symbol and the number string is greater than the distance between the numbers. Therefore, take the upper right corner of the red rectangular box, that is, the upper right corner of the RMB symbol, as the reference point, and assume that the reference point is (x, y). Through experimental analysis, select a suitable point $(x-1, y-3)$ as the starting point, extract the region of interest, and use the height of the RMB symbol template image as the height of the region to be extracted to obtain the precise digital string region as shown in Figure 11.

The amount and tax information area described above are the same, and the positions of the taxpayer identification number and invoice number in the invoice image are also unchanged. Therefore, the recognition process of the regional positioning of the taxpayer identification number and the invoice number is basically the same as the previous positioning principle. This process first directly obtains the value processing of its subregions based on prior knowledge and then uses the standard square deviation matching method to find the precise region of the number string. Different from the traditional method, this paper uses two template matching methods to extract the number string. Although the taxpayer identification number and the invoice number also have specific identifiers in front of the numeric string and they are in a fixed information area, it is inevitable that the printing is unreasonable. At this time, the position of the number string relative to the identifier will be shifted or tilted. In this case, two template matching methods are needed to extract the string. The result of extracting the information area of the taxpayer identification number in

Figure 10: Matching result map.

Figure 11: Accurate result graph.

this paper can accurately find its location, but the corresponding number string has a significant offset, and the offset location is not fixed. It may be a downward offset, or it may be offset. It is an upward shift. If the number string area is directly obtained based on prior knowledge, some data information may be lost. Here, first obtain subregion 2 based on the prior knowledge, given a wider range, so that the number string can be completely contained in the region. Then, analyze the characteristics of the number string. Both the taxpayer identification number and the invoice number are composed of more numbers, and after statistics, it is found that almost all taxpayer identification numbers have the number "1." Therefore, the number "1" is used as the template image, subarea 2 is used as the target image, and the standard square error matching method is used for matching again. The result is shown in the figure. Although subregion 2 contains Chinese characters and other noises, the structure of Chinese characters is more complicated, and the structure of Arabic numerals is quite different. Therefore, the Chinese characters in the picture do not affect the matching effect. The red rectangle is the matching result, which accurately matches "1" in the number string, which is equivalent to finding the position of the number string. Finally, according to the position of the matched coordinate point, deduct the number string in the subarea and get the accurate number string area as shown in Figures 12–16.

The main research of this paper is the recognition of uppercase amounts. The research includes a brief description of convolutional neural networks and residual networks, the preprocessing of uppercase amounts of character data, and the production of data sets, as well as a summary analysis of the test results of the two networks. This chapter combines with the previous information detection, edge detection, information identification extraction, and OCR identification to form an intelligent identification system, which automatically recognizes the reimbursement content after importing invoices. Integrating this system with the financial system of related institutions can realize intelligent financial reimbursement.

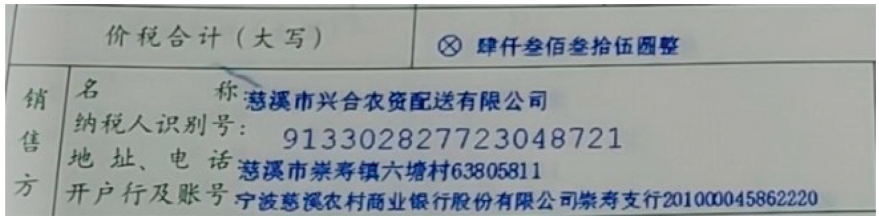

Figure 12: Small area of invoice.

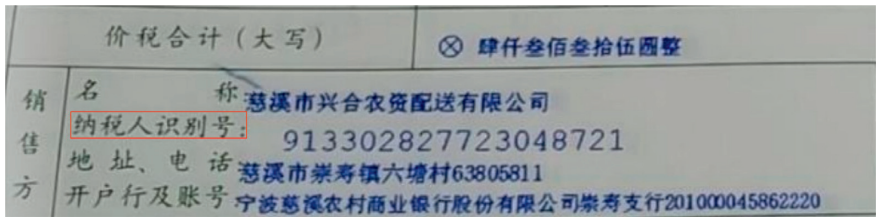

Figure 13: The first matching template.

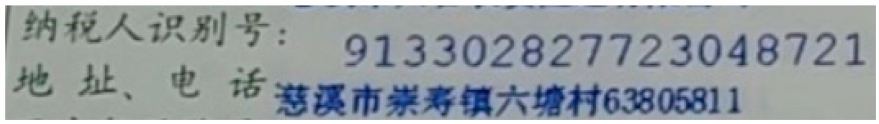

Figure 14: The second match result.

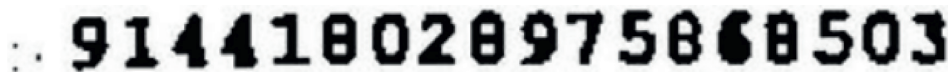

Figure 15: Test results.

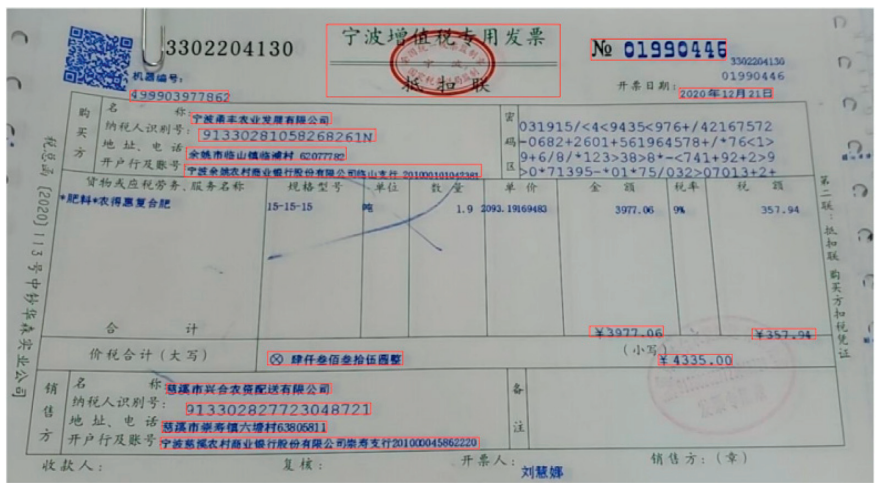

Figure 16: Overall result of invoice recognition.

4. Conclusion

With the vigorous development of artificial intelligence, the automatic invoice recognition system has also received more and more attention. At present, most of the existing recognition methods, such as Monarch Butterfly Optimization (MBO), Earthworm Optimization Algorithm (EWA), Elephant Swarm Optimization (intelligent algorithms such as EHO), and moth search (MS), are often used for image verification and recognition. Although this type of algorithm has a faster recognition rate, it is difficult to recognize problems such as invoices that have a small recognition area. Class methods generally have low recognition accuracy. This paper studies the status quo of invoice recognition and proposes an object detection and invoice recognition method based on the YOLOv3 + CRNN model. It locates the invoice information

area by marking the invoice dataset and realizes the detection and recognition of the VAT invoice information through image processing and deep learning. Finally, the system realized the rapid identification and processing of invoices. In future research, we can further optimize the information collection methods, solve subsequent data storage problems, and realize a more efficient and accurate invoice information detection and recognition system.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Smith, "An overview of the Tesseract OCR engine," *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, pp. 629–633, IEEE, Curitiba, Brazil, September 2007.

[2] A. Gonzalez, L. M. Bergasa, and J. J. Yebes, "Text detection and recognition on traffic panels from street-level imagery using visual appearance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 228–238, 2013.

[3] E. L.-C. Lai and X. Yu, "Invoicing currency in international trade: an empirical investigation and some implications for the renminbi," *The World Economy*, vol. 38, no. 1, pp. 193–229, 2014.

[4] G. Jiuxiang, W. Zhenhua, K. Jason et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[5] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[6] L. Han, C. Tianding, T. Hualiang, and J. Yingtao, "A graph-based reinforcement learning method with converged state exploration and exploitation," *Computer Modeling in Engineering and Sciences: Computer Modeling in Engineering and Sciences*, vol. 118, no. 2, pp. 253–274, 2019.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[8] 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*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[9] T. Jijun, C. Xiaolong, Z. Yingjie, and J. Lurong, "Real-time recognition and positioning of moving targets based on deep learning," *Computer Systems Applications*, vol. 27, no. 8, pp. 28–34, 2018.

[10] Z. Chaoping and Y. Yi, "Face detection and recognition in surveillance video based on YOLO2 and ResNet algorithm," *Journal of Chongqing University of Technology (Natural Science)*, vol. 8, pp. 170–175, 2018.

[11] Y. Nana, "Research on face detection algorithm based on deep learning," *Science and Technology Innovation Herald*, vol. 4, no. 26, p. 87, 2018.

[12] C. Shuhong, G. Xu, and C. Shuchun, "Moving vehicle detection based on computer vision," *Acta Metrology*, vol. 38, no. 3, pp. 288–291, 2017.

[13] H. Li, T. Chen, H. Teng, and Y. Jiang, "A graph-based reinforcement learning method with converged state exploration and exploitation," *Computer Modeling in Engineering and Sciences: Computer Modeling in Engineering and Sciences*, vol. 118, no. 2, pp. 253–274, 2019.

[14] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes: traffic flow prediction driven resource reservation for multimedia IoV with edge computing," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[15] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[16] K. Itakura and F. Hosoi, "Automatic tree detection from three-dimensional images reconstructed from 360° spherical camera using YOLO v2," *Remote Sensing*, vol. 12, no. 6, p. 988, 2020.

[17] F. Ming, S. Tengteng, and S. Zhen, "Fast helmet wearing condition detection based on improved YOLOv2," *Optics and Precision Engineering*, vol. 27, no. 5, pp. 1196–1205, 2018.

[18] N. Ganesh, R. K. Ghadai, A. K. Bhoi, K. Kalita, and X.-Z. Gao, "An intelligent predictive model-based multi-response optimization of EDM process," *CMES-Computer Modeling in Engineering & Sciences*, vol. 124, no. 2, pp. 459–476, 2020.

[19] J. Sheng, H. Min, Z. Qibing, and W. Zhenglai, "Research on pedestrian detection method based on R-FCN," *Computer Engineering and Applications*, vol. 54, no. 18, pp. 180–183, 2018.

[20] C. Rafael, N. E. Vera, J. Lucas, and F. V. Ferran, "An ETD method for American options under the heston model," *CMES-Computer Modeling in Engineering & Sciences*, vol. 124, no. 2, pp. 493–508, 2020.

[21] L. Cen, G. Lijun, Z. Rong, and H. Yetian, "Application of improved YOLOv3 algorithm in container number positioning," *Sensors and Microsystems*, vol. 46, no. 7, 2019.

[22] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.