Complex Labels Text Detection Algorithm Based on Improved YOLOv5

Yingning Gao, Weisheng Liu

Abstract—**Complex labels have been widely used in various industries. The accuracy of its content is critical both in the fields of people's livelihood, such as supermarkets and shopping centers, and in the management of goods in the industrial fields, such as logistics and factories. Inaccurate label information identification can make item management difficult. Because complex labels can simultaneously contain text, icons, bar codes, QR codes, and other information with different aspect ratios. Traditional methods like feature extraction and template matching have problems, such as detection frames breaking between Chinese, English, and numeric symbols. As a result, entire lines of text on complex labels cannot be detectable, resulting in low detection accuracy. In this paper, a deep learning-based text detection algorithm was proposed. By employing the operation of inverse convolution, improved the object detection algorithm you only look once 5 (YOLOv5). In the backbone part of the original model, involution is used instead of the convolution layer to improve target classification and prediction. The original anchor frame was modified using *k*-means clustering to make it more applicable to text of various sizes in labels. The enhanced algorithm is called as Invo-YOLOv5. Experiments show that this model can significantly improve detection efficiency while also addresse the problems of false detection and missed detection. Finally, the detected text is verified by using CRNN and Tesseract OCR with complex labels as samples for recognition. Both methods can be effectively recognized, demonstrating the efficacy and generality of the Invo-YOLOv5 method in the process of complex labels text detection and improving the detection accuracy.**

Index Terms—**Complex Labels, Involution, Text Positioning, YOLOv5**

I. Introduction

With the development of the social economy, the label industry is evolving in the direction of personalization and refinement. Meanwhile, the labeling application environment is becoming more diverse, and the industries and scenarios requiring labeling are expanding. A complex label can include icons, bar codes, QR codes, and text in different aspect ratios, among other things. Using primitive OCR text recognition technology to handle such complex labels will suffer from low detection accuracy, complex procedures, and time-consuming problems. In the text detection, traditional methods commonly used are divided into the following two types.

- Feature extraction [1], input classifier, and OCR model generation [2]. On the one hand, this method extracts features such as lines, edge lines, and intersections from character structure, and the extraction results have a direct impact on the final detection results. On the other hand, classifiers are used to categorize artificially designed character features. This method requires a significant amount of manpower and time. And the model is generalization ability is reduced when the problems such as fuzzy characters, distortion, complex backgrounds, or other noise effects are encountered.
- Classical character template matching method. In 2020, Susan *et al*. proposed an adaptive threshold texture detection method to locate and segment text areas from document images [3]. That is using the OTSU threshold to binarize the text, and improving the image texture by the sliding window. However, due to the complexity of the background, the template matching method can only be used for some simple situation.

To address these problems, we proposed an intelligent text detection method. This method is an optimized object detection algorithm that can directly locate the text in the input image and output the category information. This method can well adapt to different character sizes and categories in complex labels. The core of the algorithm is as follows:

- Text detection: Analyze the structure of the input image and locate the position of the text in the image. Provide input for next-step identification.
- Text recognition: Convert the text information of the input image into a string and output the result.

Text detection task is not only the foundation of the whole system, but also the most critical step. Accurate positioning can ensure that subsequent identification proceeds smoothly. This paper focuses on the text detection algorithm and proposes solutions to problems encountered during the text detection stage.

The biggest challenge in applying the target detection model to detect text with complex labels is that the labels usually contain characters of multiple categories, such as Chinese, English and numbers. There are also a lot of interference information, such as trademarks, bar codes and symbols, which complicates text detection. Furthermore, different types of text spacing and string length are also different. Above are the main causes of detection information loss.

YOLOv5 is a common method in target detection. This

Manuscript received May 31, 2022; revised February 15, 2023. This work was supported by the Special Fund for Scientific Research Construction of University of Science and Technology Liaoning.

Yingning Gao is a postgraduate student at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: gyningg@163.com).

Weisheng Liu is a professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author: tel:+086-13942297366; fax: 0412-5929809; e-mail: succman@163.com).

method improves the speed of label detection in industrial situations to some extent by using text region instead of the single character stitching method for text detection. Although this single-phase detection technique has certain speed advantages, its detection accuracy is not satisfactory because it simply uses a CNN network to forecast various target categories and locations. Based on this, this paper have proposed an improved YOLOv5 algorithm Invo-YOLOv5 to detect text in complex labels. Specifically, we have modified the original anchor boxes to accommodate text content with different aspect ratios. In addition, because convolution is the foundation of neural networks, the backbone part of YOLOv5 is primarily stacked by residual convolution, and convolution has a direct impact on the model is final prediction and classification results. In particular, the spatial invariance and channel specificity of the convolution kernel. The spatial concentration of the receptive field into one channel dimension affects the ability of the convolution kernel to extract features flexibly in visual tasks. Therefore, this paper modified the original backbone network and build a new backbone feature extraction network called Invo-CSPDarknet. The newly designed convolution is stacked with the original convolution and updated using a backpropagation algorithm. The convolution kernel learned in this way can adapt to images of different resolutions, resulting in the more intuitive and effective detection results. Invo-CSPDarknet reduces redundancy by sharing the convolution kernel along the channel dimension. The designed convolution can be embedded in the detection algorithm in a lightweight way to facilitate target recognition.

The improved YOLOv5 model has two characteristics: (1) It breaks through the limitation of the original space range and can detect content in a wider space. (2) Adaptive weight allocation at different positions is conducive to the extraction of optimal element information. In the third and fourth sections, the algorithm structure and experimental results will be discussed.

II. Related Work

In the field of text region detection, the classical classification methods can be divided into edge-based and texture-based detection methods. For example, in 2010, Neumann *et al*. proposed applying the MSER (maximally stable extremal regions) algorithm to natural scene text detection. Text candidates have been obtained by detecting maximally stable extreme regions in the image to add robustness to geometry and lighting conditions [4]. In 2018, Yang *et al*. proposed a FASTroke key point extractor based on MSER to detect Uyghur text in complex backgrounds. Cluster analysis of component similarity was presented by incorporating component-level classifiers to reduce computational costs [5]. However, the Uyghur text in the datas usually has uniform features (size, color, and anchor width). Once the text information has a large feature difference, the detection effect will be greatly reduced. The conventional detection techniques described above are constrained by fixed patterns, which require extensive manual setup and a priori knowledge to distinguish the textual part of the image. Moreover, errors will increase with the increase of detection steps, affecting the detection effect, so it is not suitable for complex labels text detection. In recent years, deep learning has developed rapidly and is widely used in the field of text detection to improve algorithm performance. The representative method is based on semantic segmentation [6] and object detection [7].

The detection algorithm based on semantic segmentation analyzes each pixel in the image to determine whether it belongs to the text area and then carries out pixel fusion of the text area to get the final text area. In this regard, in 2018, Deng *et al*. proposed a scene text detection algorithm, PixelLink, to extract text locations directly from strength segmentation results, and deep neural networks (DNNs) were trained to perform two types of predictions, namely text/nontext prediction and link prediction, where positive pixels belonging to text are linked together by positive links to obtain text regions [8]. In 2019, Xie *et al*. proposed a supervised pyramid context network (SPCNET) to locate text. Inspired by Mask R-CNN, this network uses semantic segmentation to assist detection branches to capture context information and improve algorithm performance [9]. In 2019, Scholars Baek *et al*. proposed the character-level text detection method character region awareness for text detection (CRAFT), which first detects individual characters and the connection relations between characters, and then determines the final text line based on the connection relations [10]. However, since the internal adhesion and feature differences of images were large, it was difficult to distinguish them using the semantic segmentation method of text/nontext.

The other detection algorithm, which is based on target detection, outputs the text position directly. For example, Liao *et al*. proposed TextBoxes++ [11], which was based on the target detection algorithm SSD [12], combined with the nonmaximal suppression (NMS) method to detect text regions. Based on the multi-stage target detection algorithm Faster R-CNN [13], Elanwar *et al*. proposed an FFRA page layout analysis structure to locate Arabic text regions in books [14]. Cao *et al*. combined CTPN and the target detection algorithm YOLOV3 for text detection of student exercise images [15]-[16], improved the network by changing the regression object from a single character to a fixed-width text and performing fractional threshold filtering and nonmaximum suppression filtering on the output boundary box to improve the detection effect. Such target detection algorithms are applied to the text detection field, which needs to be modified according to the characteristics of the image data to design a more suitable anchor box.

Based on the object detection algorithm YOLOv5, this paper has constructed a complex labels text detector. Experiments show that compared with the text detection algorithm that uses a single character as the regression object, the algorithm in this paper has higher accuracy. In addition, by inputting the detected text into the basic recognition model CRNN and Tesseract-OCR, the text could be effectively recognized, which proves the validity and universality of the model.

III. Methodology

As an application example of OCR technology, complex labels text recognition system plays an important role in industrial production. The whole system consists of three parts: image preprocess, text position detection, and text recognition. The improved YOLOv5 text detection algorithm was combined with the text recognition algorithm, which conforms to the construction idea of modern intelligent work and could adapt to more complex application environments.

The whole algorithm flow is shown in Fig. 1. Firstly, the original image samples were preprocessed to change the definition, and expanded to increase the diversity. Then it was sent to the improved YOLOv5 network for training, and the label text detection model Invo-YOLOv5 was obtained. The detection model located the text information in the sample sends it to the text recognition network for training, and finally got the end-to-end complex labels text recognition system.

A. Model Structure and Optimization

1) The Original YOLOv5 Model Structure

YOLO series networks are typical one-stage object detection algorithms that directly regress the location and category of objects in the output to complete fast detection [16]-[19]. Among them, YOLOv5, the latest achievement of the YOLO series, has achieved a good test effect in the COCO data sets. Compared to previous versions, YOLOv5 uses CSPDarknet53 as the backbone feature extraction network, which includes three modules: CBL, bottleneckCSP and SPP. The CBL structure consists of a three-layer network of convolution, batch standardization, and activation functions. The bottleneckCSP architecture consists of a 1×1 standard convolution stacked with 3×3 residual convolutions that reduce the number of parameters. SPP is a spatial pyramid pool structure, which can extract features from the feature map from different angles. Three effective feature layers are obtained by backbone for regression prediction in the neck and YOLO head. Therefore, the construction of a backbone feature extraction network is particularly important [20].

The YOLOv5 algorithm has been used to detect larger objects. If it is used to detect text directly, there will be some detection accidents, because the text has unique characteristics:

- The original anchor box does not work with text. The text is presented in the form of a long rectangle, whereas the aspect ratio of common object detection models is almost 1.
- Confuse text and background. The text boundary has no obvious closed outline.
- The detection frame is broken. Complex labels are rich in information and often contain multiple characters with different length-to-width ratios and intervals.

To make the model more suitable for complex labels text detection tasks, we modified the original YOLOv5 as follows.

2) Improved Invo-YOLOv5 Algorithm

Based on classical image filtering methods, the convolution kernel has two characteristics: spatial invariance and channel specificity [21]-[23]. Thanks to these two properties, the convolution kernel can move in the same channel space without changing the size of the kernel. The convolution kernel in the YOLOv5 model has two sizes of 1×1 and 3×3, and the width of the number of channels in the feature graph increases with the increase of the number of network layers. But for neural networks, the channel number is not as wide as possible. If the convolution kernel is only active between feature graphs with the same number of channels, the weight adjustment parameters cannot be updated flexibly.

To solve the above problems, this paper improved the structure of a specific part of the convolutional network of the YOLOv5 model. The properties of space and channel had been interchanged, involution had been introduced to replace the convolutional layers in the CBL structure of the original model [23], and involution had been added between the 1×1 and 3×3 convolutions of the bottleneckCSP standard. The cross-channel feature extraction can improve the target detection accuracy to a certain extent. In the label text detection task, the average precision mAP increased from 90.06% to 92.36%. Fig. 2 shows the structure diagram of the improved Invo-YOLOv5 network. The main experimental operations are as follows:

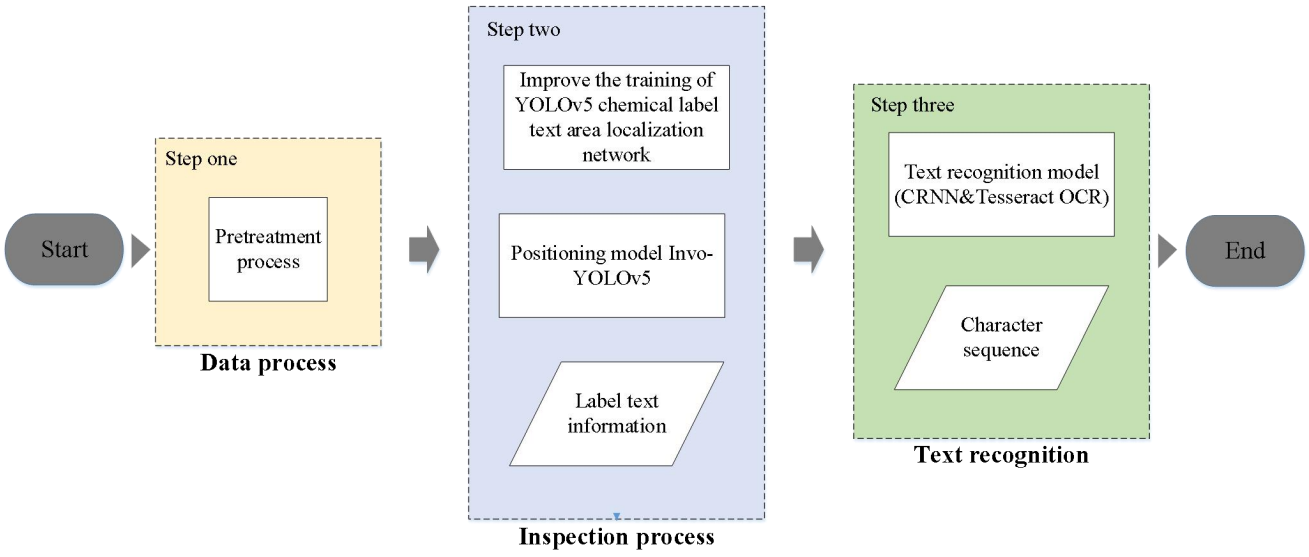

Fig. 1. Flowchart of the entire algorithm

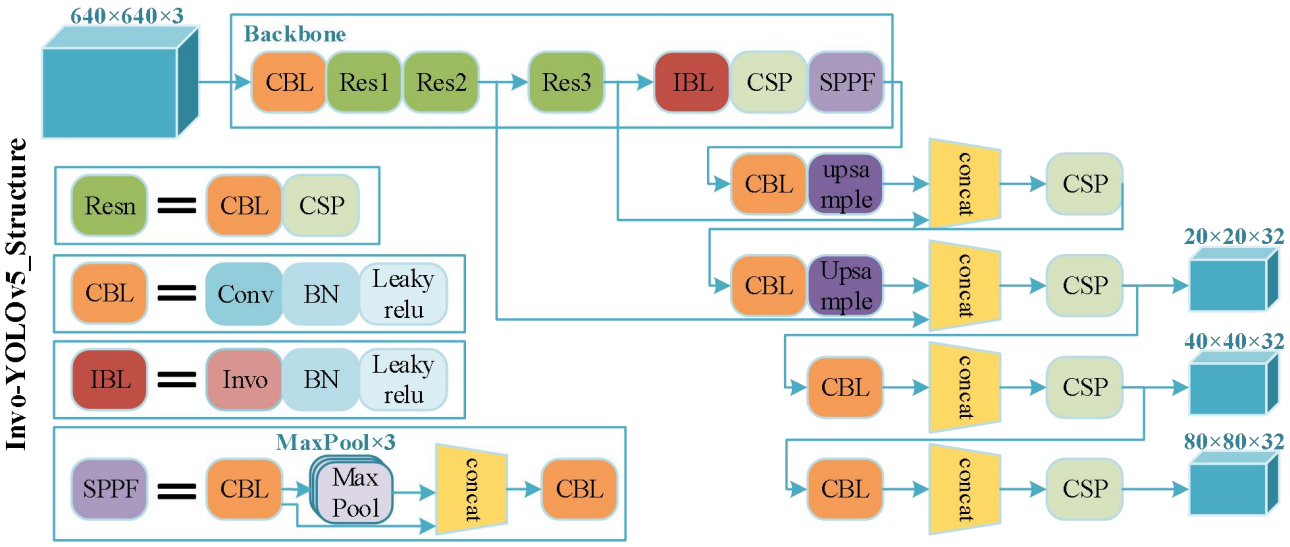

Fig. 2. Backbone structure of the modified Invo-YOLOv5 model

- Modified the anchor frame settings. Adapted text lines with a large difference in aspect ratio.
- Modified the original convolution. The backbone CBL part, whose the channel number was 1024 layers of convolution, was modified to 512 involution. The involution was introduced into the bottleneck CSP structure, and the high semantic features were extracted. The model parameters were reduced, and the computational complexity was reduced by more than half.

a. Candidate Frame Design Based on A Complex Labels Dataset

During the experiment, we applied different object detection algorithms to label text detection. It was found that most detection algorithms are unable to complete text lines, with problems such as overlapping, misaligned and broken bounding boxes. Therefore, we analyzed the anchor box setting mechanism of the algorithm and designed an algorithm that was more suitable for different forms of text detection within complex labels.

In order to accurately locate the target position, YOLO would set the initial length and width of the anchor frame. These anchor frames of different sizes and proportions enable each grid to output the prediction frame, compared with the real frame, calculate the gap, reverse the update, and finally iterate the network parameters. The setting of the box would affect the final detection accuracy. If the recall rate of the annotation information of the data set to the default anchor box was less than 0.5 calculated before training, *k*-means would be used to analyze the location annotation information of the labels in the training set. The optimal preset anchor frame was recalculated to better fit the size of the target to be detected, and improve the convergence speed of the model. Suppose the target was divided into k clusters, and the k-means clustering algorithm uses the error squared and E criterion functions to evaluate the performance. Set the data set X contains k clustering subsets X_1, X_2,..., X_k, and the cluster centers are m_1, m_2,..., m_k respectively. Then the sum of squared errors of the objective function is as follows:

$$E = \sum_{i=1}^{k} \sum_{p \in X_i} \|p - m_i\|^2 \quad (1)$$

The calculation process was as follows: (1) Initialize k points randomly as clustered cluster centers; (2) Calculated the distance between each sample point and each cluster center and assigned the closest cluster; (3) Uses the mean of each cluster sample to update the cluster center; (4) repeated (2) and (3) until the clustering center no longer changed.

The dimensions of the 9 candidate frames were obtained by clustering: (48×14), (100×17), (162×14), (76×33), (158×22), (228×15), (344×15), (216×26), (317×32), the newly obtained anchor frames were used for the YOLO model, with each output feature layer corresponding to 3 candidate frames. The aspect ratio was calculated directly using the real frame compared to the candidate frames, for any of the ground truth.

The convolutional neural network divides each feature map into many units, as shown in Fig. 3. The dimension of the original image after preprocessing were 640×640×3. After passing the convolutional neural network, the output dimension of the detection layer of three scales is $S \times S \times n_a \times (t_x + t_y + t_w + t_n + t_o + n_c)$, where $S \times S$ is the number of divided grids, n_a is the number of preset boxes corresponding to each scale, and n_c is the number of categories to be detected.

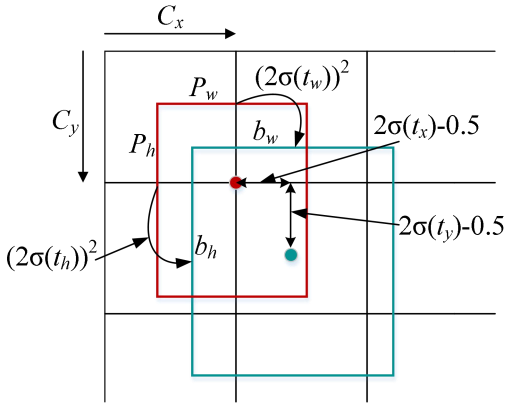

Fig. 3. YOLOv5 prediction bounding box decoding schematic

In addition, the convolutional neural network predicts four values for each bounding box on each cell. The offsets of the bounding box relative to the upper-left corner of the cell are t_x and t_y, and the scales relative to the anchor boxes are t_w and t_h. Assuming that the cell offset relative to the upper left corner of the image is (c_x, c_y) and the anchor box height and width are p_h, p_w. Then the coordinates of the target prediction bounding box are (b_x, b_y) the width is b_w, and the height is b_h. The calculation method is as follows:

$$\begin{cases} b_x = 2\sigma(t_x) - 0.5 + c_x \\ b_y = 2\sigma(t_y) - 0.5 + c_y \\ b_w = p_w\left(2\sigma(t_w)\right)^2 \\ b_h = p_h\left(2\sigma(t_h)\right)^2 \end{cases} \tag{2}$$

b. Invo-YOLOv5 Backbone

YOLOv5 uses CSPDarknet53 as the backbone to extract rich information features from the input image. The extracted features are called the effective feature layer and serve as the feature set of the input image. The backbone of YOLOv5 was composed of ordinary convolution kernel residual convolution, and we modified it on this basis. The modified module can be well integrated into the original algorithm.

a) Principle of Convolution

Convolutional layers, also known as filters, were designed to extract different features from input data. The first convolution layer would be used to extracted some lower-level features, such as edges, lines, and corners. With the deepening of the network layers, the extracted features would become more complex. Convolution kernel parameters were usually 4, including the convolution kernel size, step size, edge filling method, and a number of output channels. The working principle was a two-dimensional matrix involved in the operation, at a certain step on the input image translation, dot product operation, to get the output matrix. The original convolution is calculated as follows:

$$Y_{i,j,k} = \in \sum_{c=1}^{C_i} \sum_{(u,v)\in\Delta_k} F_{k,c,u+[K/2],v+[K/2]} X_{i+u,j+v,c} \tag{3}$$

TABLE I
INVO-CSPDARKNET IMPLEM ENTATION CODE

```
# b: batch size, h: height, w: weight, c: channel number
################### initialization ###################
if stride > 1:
self.avgpool = nn.AvgPool2d(stride, stride)
self.unfold = nn.Unfold(kernel_size, 1, (kernel_size-1)//2, stride)
#################### forward pass ####################
weight = self.conv2(self.conv1(x if self.stride == 1 else
self.avgpool(x)))
b, c, h, w = weight.shape
weight = weight.view(b, self.groups, self.kernel_size**2, h,
w).unsqueeze(2)
out = self.unfold(x).view(b, self.groups, self.group_channels,
self.kernel_size**2, h, w)
out = (weight * out).sum(dim=3).view(b, self.c1, h, w)
return out
```

Where, $X \in \mathbb{R}^{H\times W\times Ci}$ is the input feature graph, H, W, and C_i respectively represent the height, width, and number of input channels. The featured graph can be regarded as several cells with features. The convolution kernel C_o has a fixed size $K\times K$, and the set of convolution kernels is denoted by $F \in \mathbb{R}^{Co\times Ci\times K\times K}$. When the convolutional network recognizes the information input, it transforms the image into a sliding window to produce the output feature map. $\Delta\kappa \in \mathbb{Z}^2$ represents the set of center offsets of adjacent pixel points.

b) Invo-CSPDarknnet Network Design

A total of four convolution structures were stacked in the YOLOv5 trunk feature extraction network. For classification and dense prediction, involution is used to replace the 3×3 convolution in the last stack block of CSPDarknet53. The original CBL structure was changed into the IBL structure, and the work of convolution standardization plus activation function was completed. The 1×1 convolution in the channel was left to continue the work of linear operations. At the same time, an involution layer was added between the two standard convolutions of the bottleneck. It alleviates the problem of gradient disappearance with the increase of depth and reduces computational complexity. The formula is as follows:

$$Y_{i,j,k} = \sum_{(u,v)\in\Delta_K} P_{i,j,u+[K/2],v+[K/2],[kG/C]} X_{i+u,j+v,k} \tag{4}$$

Where, $P \in \mathbb{R}^{H\times W\times K\times K\times G}$ represents the new kernel with the opposite property to the original, and G in the kernel was used to calculate the number of groups of shared convolution kernels. Similarly, the effective feature layer was still obtained in the form of a sliding window. The shape of kernel P is determined by the input feature map. The involution kernel demo is shown in Fig. 4. In the picture, $G = 1$ is used as an example, the convolution kernel P was generated at the function Φ, and $\otimes$ represents multiplication across channels and summation of aggregates in the neighborhood of $\oplus$ space. The generation function of kernel P is as follows:

$$P_{i,j} = \emptyset\left(X_{i,j}\right) = W_1\sigma\left(W_0 X_{i,j}\right) \tag{5}$$

In this formulation, there are two linear operations, W_0 and W_1, which together form the head structure of the network. A denotes batch normalization. The intermediate channel dimension is constrained by the nonlinear activation functions of the two linear projections. The pseudocode in Table I describes the specific deployment process for Invo-CSPDarknet.

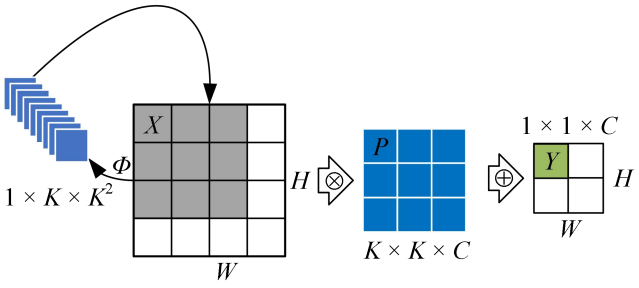

Fig. 4. Involution kernel demonstration diagram

In the experiment, Invo-CSPDarknet was well embedded into the original model in the form of a stack, and the newly formed text detection model was called Invo-YOLOv5. The backbone process was as follows: First of all, the input feature map would be scaled to the size of $(640 \times 640 \times 3)$. After two original convolutions, the size would be converted to $320 \times 320 \times 64$. Then, three original stacked convolution CBL modules would be entered. In the final residual convolution module, a 3×3 involution and an original 1×1 convolution intersperse the exchange of channel information gathering information in the rich receiving domain. After optimizing convolution processing, the number of feature graph channels would be changed from 1024 to 512. This operation effectively reduces the amount of computation and improves the model performance. Finally, the backbone obtains three effective feature maps, the sizes of which were $(80 \times 80 \times 256)$, $(40 \times 40 \times 512)$, and $(20 \times 20 \times 512)$. For subsequent feature enhancement and result prediction.

B. Text Recognition Algorithm

1) CRNN Model Structure

CRNN is an end-to-end training model. Since word recognition is a predictive method for sequences, CRNN adopts an RNN network. First, the features of the model were extracted by CNN, then the sequences were predicted by RNN, and finally the final results were obtained by a CTC transcription.

The gradient disappearance problem in traditional RNN networks limits the range of contexts it can store. In response, CRNN introduced bidirectional long short-term memory (LSTM) [24]. As a variant of the temporal recurrent neural network, this special design guarantees the extraction of sequence features.

The main works are divided into four steps: feature extraction, sequence transformation, executing LSTM to obtain sequence output, and performing CTC transformation. The combination of local feature information and sequence information was realized, which could better describe the data feature information. In this paper, for complex labels text recognition, the specific processes are as follows:

- Cropped datas. For the text localized by the previous target detection, it was uniformly cropped to make it suitable for the CRNN model requirements.
- Extended training picture. Due to the small amount of training data, it was difficult for the model to converge. So this paper expanded the data sets by randomly disrupting the training data images, and finally got 19266 training sets and 599 test sets. The input image height was 32 times the downsampling, and the width comes according to the actual situation.
- Feature extraction. The convolutional layer uses a 7-layer convolutional neural network, and the core part of the network is shown in Table II. Improved from the VGG network, the last two layers of maximum pooling layer window size are changed from 2×2 to 1×2 to make it more adaptable to the detection of long text. A batch normalization (BN) layer was added after the convolutional layers of layers 5 and 6 to normalize the input data, accelerate the model convergence.
- Sequence transformation. The recurrent layer, composed of a bidirectional recurrent neural network, predicts the feature sequence obtained from the convolutional layer and obtains the corresponding label distribution. The bidirectional two-level long-memory network (BLSTM) was introduced to solve the problem of gradient disappearance or explosion caused by too-long sequences.
- CTC transcription. The predictions made by the RNN for each feature vector were converted into sequences, and the sequence of labels with the highest probability combination is found based on each frame prediction. The end-to-end joint training of CNN and RNN using the loss function of CTC (Connectionist Temporal Classification, CTC) introduces blank characters to solve the recognition challenge of OCR and how to handle indeterminate long sequence alignment. The final partial recognition results are shown in Part IV.

2) Tesseract-OCR

Tesseract Optical Character Recognition is an open-source OCR engine that supports text detection in multiple languages and is divided into two parts in general: image layout analysis and character recognition. The training processes are as follows:

- Captured the image library. Fusing all complex labeled training library into .tif format.
- Generated and adjusted character borders. Combine the images generated in the first step into one and make sure the characters inside each box is complete.
- Defined the character configuration file and generated the .box file.
- Generated training and character set files.
- Merged all the training files generated in the previous steps. Produced new libraries of complex labeled text languages.

TABLE II
CRNN MODEL BACKBONE NETWORK

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512,k:2×2,s:1,p:0
MaxPooling	Window:1×2,s:2
BatchNormalization	-
Convolution	#maps:512,k:3×3,s:1,p:1
BatchNormalization	-
Convolution	#maps:512,k:3×3,s:1,p:1
MaxPooling	Window:1×2,s:2
Convolution	#maps:256,k:3×3,s:1,p:1
Convolution	#maps:256,k:3×3,s:1,p:1
MaxPooling	Window:2×2,s:2
Convolution	#maps:128,k:3×3,s:1,p:1
MaxPooling	Window:2×2,s:2
Convolution	#maps:64,k:3×3,s:1,p:1
Input	W×32 gray-scale image

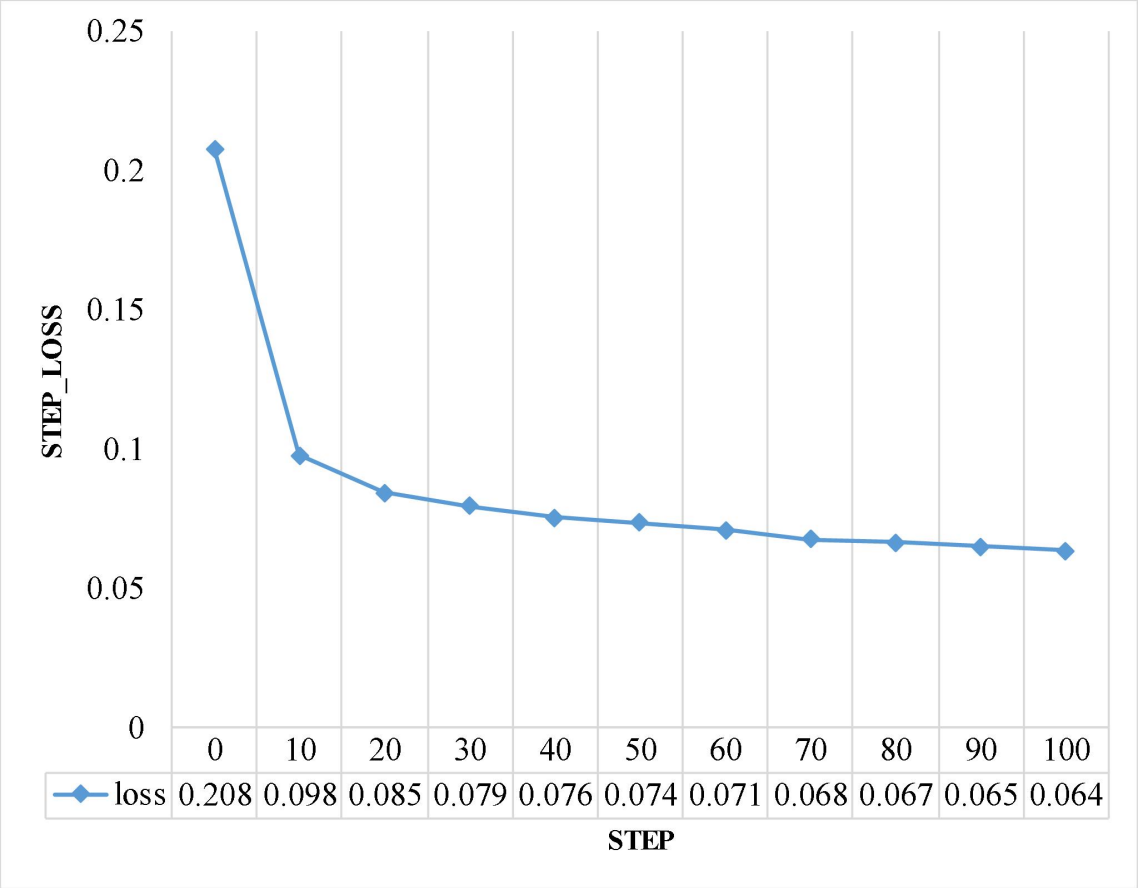

Fig. 5. Train loss curve

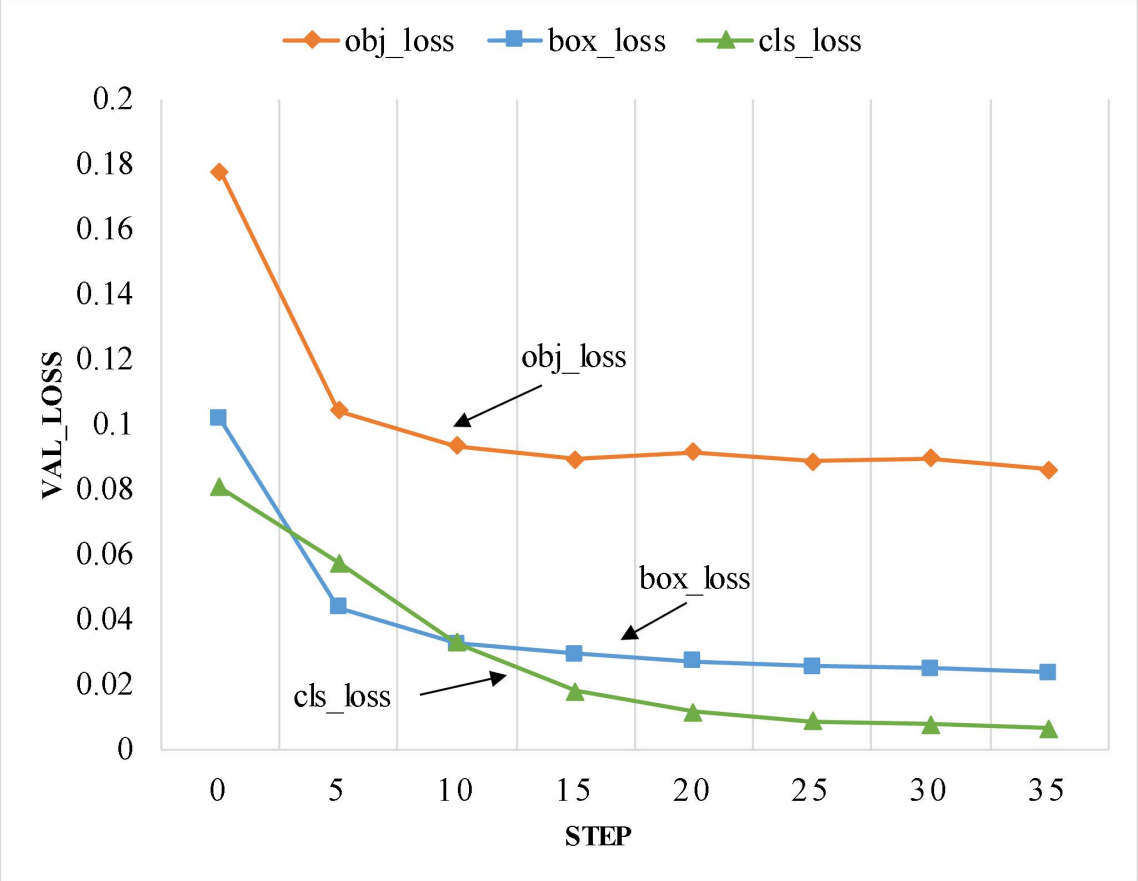

Fig. 6. Test loss curve

IV. Experiments

A. Experimental Dataset of Complex Labeled Images

In this study, we used the complex labels collected by the scanner as the experimental data sets. Each label contains information such as numbers, English and Chinese, bar codes, QR codes, and images. LabelImg was used for marking, and the jpg file was converted to an xml file conforming to the standard. Before the experiment, the data of the picture had been enhanced to generalized the training data and reduce the over-fitting situation. There are two main forms of data enhancement:

- Make geometric changes based on training images to improve the robustness of the model.
- Sharpening enhancement makes the text clear. The result was 4,946 images, 4,472 of which were used for training and the rest for testing.

The loss function is an important evaluation basis for optimization and updating of model parameters. For the algorithm in this paper, the loss function of text detection was composed of three parts: boundary box regression loss, confidence prediction loss, and category prediction loss. YOLOv5 used *CIoUloss* to calculate the bounding box regression loss function $loss_{box}$, where $loss_{obj}$ was the judgment of network confidence and $loss_{cls}$ was the judgment of the category of objects contained in the anchor frame. The classification loss in the training stage adopts the binary cross-entropy loss. The specific formulas are as follows:

$$Loss = loss_{box} + loss_{obj} + loss_{cls} \tag{6}$$

$$Loss_{box} = CIoUloss = 1 - CIoU \tag{7}$$

$$CIoU = IoU - \frac{|A_c - U|}{|A_c|} \tag{8}$$

Where *IoU* is the intersection ratio between the real box and the predicted box. Subtract the minimum closure area of the two boxes from the value of *IoU* (the minimum area of the box that contains both the predicted and the real box), and *CIoU* gets the function value.

B. Implementation Details

The epoch of the network was set to 100. In order to avoid overfitting in the parameter adjustment process, the initial learning rate was set at 0.01, and the weight attenuation of the learning rate was set at 0.0005. In the process of training gradient optimization, the learning speed might be too slow, so the momentum was set to 0.9 to speed up the learning speed, and the SGD stochastic gradient descent optimizer was used. The specific parameters are shown in Table III.

As can be seen from Fig. 5, the loss decreases rapidly within 10 rounds, and then the downward trend gradually decreases and becomes gentle, and the loss decreases from 0.1952 to 0.06391 in the whole training process. The results show that the training process is effective.

Then the trained model is evaluated on the test set, and the results are shown in Fig. 6. Three loss curves, cls_loss of category, box_loss of bounding box, and obj_loss of confidence, are used to reflect the training effect. It can be seen that the bounding box and the confidence loss ratio class loss converge first. It gradually converged after 5 epochs, and the category loss also leveled off after 15 epochs, eventually remaining around 0.86.

C. Experimental Results and Analysis

1) Evaluation Indicators

We tested five models on the same complex label datasets, namely the Invo-YOLOv5 model, the original YOLOv5, and three other classic target detection algorithms, YOLOv4 tiny, YOLOx, and SSD.

In the object detection task, *TP* (true positive), *TN* (true negative), *FP* (false positive), and *FN* (false positive) are used to calculate specific evaluation indicators. *Precision* reflects the number of categories detected that truly belong to that category. The formula is as follows:

$$Precision = TP/(TP + FP) \tag{9}$$

Recall represents the proportion of a certain category of target detected, as a percentage of the real box. The formula is as follows:

$$Recall = TP/(TP + FN) \tag{10}$$

F1 represents the proportion of the number of false detection targets detected by the detection algorithm. The formula is as follows:

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{S + TP - TN} \tag{11}$$

AP is a single-sample accuracy value using *P* (precision) and *r* (recall). where *n* is the number of categories. The formula is as follows:

$$AP = \left(\frac{1}{n} \sum_{\left(r \in \frac{1}{n}\frac{2}{n} \cdots \frac{n-1}{n}, 1\right)} P_{\text{interpo}(r)} \right) \tag{12}$$

The average precision *mAP* is the *AP* average of all detected classes. The calculation method is as follows:

$$mAP = \left(\frac{1}{n} \sum AP \right) \tag{13}$$

TABLE III
INVO-YOLOv5 TRAINING PARAMETERS

Training parameters	Numeric value
Epochs	100
Batch-size	64
Learning 0	0.01
Weight_decay	0.0005
Momentum	0.9
Training set	4472
Test set	497

2) Results and Analysis

The experimental results show that, compared with other detection models, our Invo-YOLOv5 model has a better detection effect. The main reason for the improved performance was the use of an improved anchored box to obtain a check box with higher text confidence. Moreover, involution was added to the backbone part, which prompts the model to extract rich semantic features between different channels and reduces the computational complexity of the YOLOv5 model. Compared with other common target detection models, the average accuracy is improved by 2.3%, 0.53%, 3.16%, and 2.44%, respectively. The results and specific analysis are shown in Table IV.

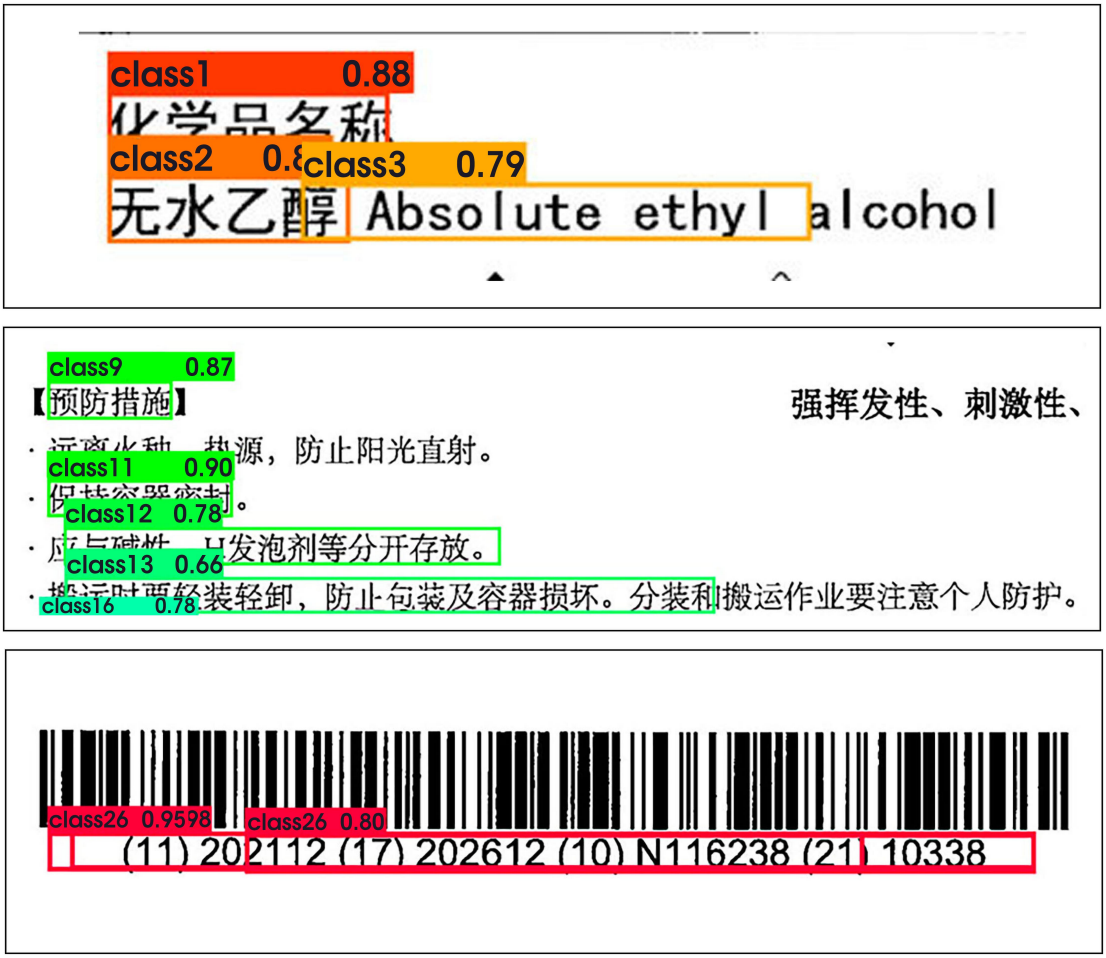

Fig. 7 Detection effect of other algorithms

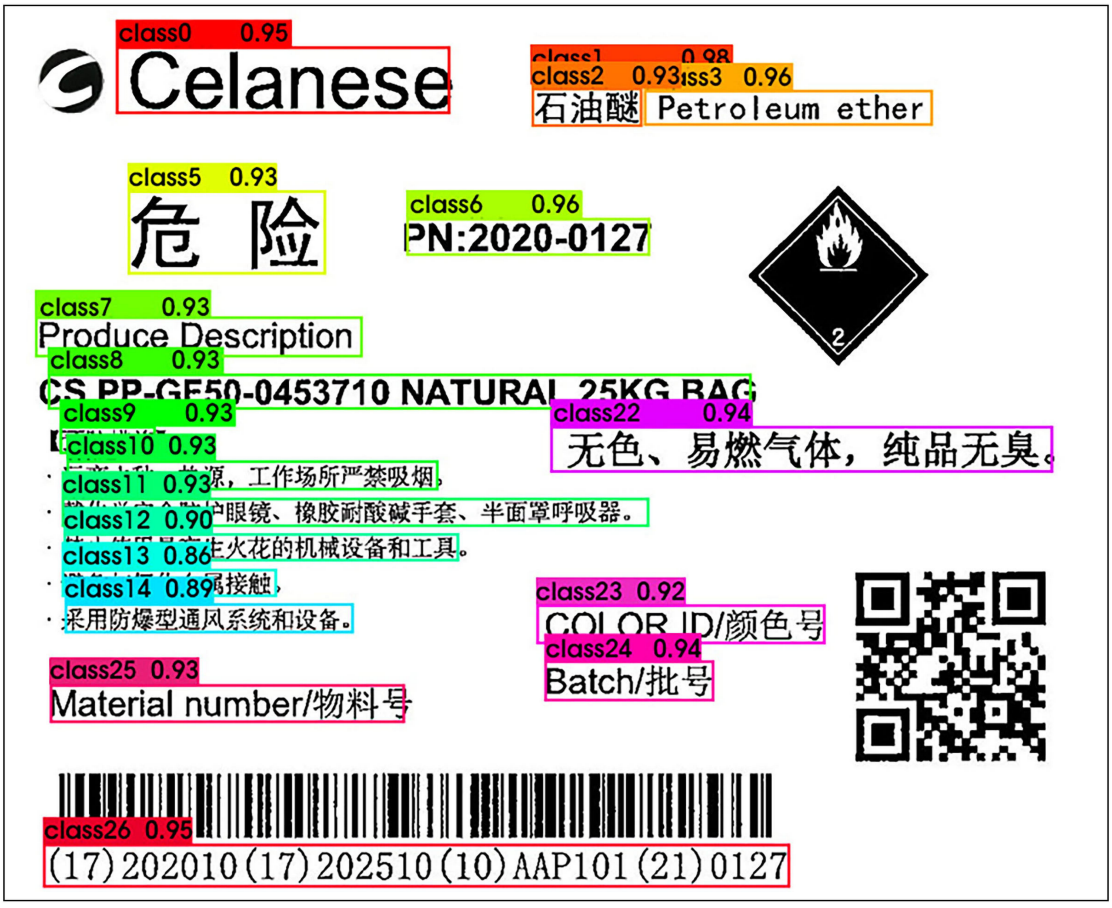

Fig. 8. Model test rendering of Invo-YOLOv5

TABLE IV
MODEL TRAINING RESULTS

Method	Backbone network	mAP(%)
Invo-YOLOv5	Invo-CSPDarkNet	**92.36**
YOLOv5	CSPDarkNet	90.06
YOLOx	CSPDarkNet	91.83
YOLOv4-tiny	CSPDarkNet53-tiny	89.20
SSD	VGG16	89.92

All models in the experiment were one-stage object detection methods. The main idea was to carried out intensive sampling evenly in different positions of the input image after using CNN to extract features. Prediction boxes of different scales could be used during sampling to realize simultaneous classification and prediction. Among them, YOLOx is based on the benchmark model YOLOv3-spp and was obtained through a series of improvements. Yolov4-tiny is a simplified version of YOLOv4 that uses only two feature layers for classification and regression prediction. SSD is also a one-stage general object detection algorithm. The backbone network is VGG16. Its core function is to predict object as well as the score of belonging categories. All the other models except SSD have an FPN neck, which is part of the feature fusion of the effective feature layer obtained from the trunk.

All experiments of text detection were first pretrained on PASCAL VOC2007 data sets, and finally adjusted on this experimental data sets. The epoch of all models was set to 100 and the initial learning rate was 0.01. In the experiment, we found that there was no significant difference in the detection effect of the above algorithm for conventional hor-izontal text. However, when the line of text contains both Chinese and English, numbers, and symbols, other algorithms could be problematic. This is shown in Fig. 7. It can be seen that the original YOLOv5, YOLOx, YOLOv4-tiny, and SSD exist problems: (1) Long text detection is incomplete. (2) The text detection box overlaps.(3) The detection box breaks between different types of characters. Our improved model can effectively solve the above problems, as shown in Fig. 8.

D. Text Recognition Verification

The ultimate goal of our algorithm is to recognize the detected text, and two different recognition algorithms are used to verify the proposed text detection algorithm. Here, this paper used the basic CRNN model and Tesseract OCR to identify the located text information. There were 19,266 training sets and 599 test sets, and the final CRNN recognition accuracy was 78.95%. Fig. 9 shows the label text recognition results, including long text of safety measures, product descriptions composed of special symbols of Chinese characters, and long code number information composed of digits.

The experimental results show that the network model based on improved YOLOv5+CRNN/Tesseract-OCR has significant advantages in text detection and recognition of complex labels.

It can be seen from the experimental results that the network model based on the improved YOLOv5+CRNN/Tesseract OCR has superior performance in the task of complex labels text detection and recognition.

(a) Original image

(b) Identification results

Fig. 9. CRNN identifies the result graph

V. CONCLUSION

This paper focused on text detection based on complex labels, proposed a new text detection algorithm, and improved the target detection network YOLOv5 to improve the accuracy of text detection. This algorithm had two main parts. Firstly, the anchor frame was improved by the image text feature. Secondly, a new backbone network, Invo-CSPDark Net, was proposed to make the algorithm more suitable for text lines with different aspect ratios. Compared with the original YOLOv5 network, the accuracy of our algorithm was improved by 2.3%. Finally, the text accurately detected was provided to subsequent CRNN and Tesseract OCR text recognition algorithms to prove the universality of the detection algorithm.

Although the existing algorithm can satisfy most application scenarios, we still need to improve in some areas. First of all, our algorithm is mainly aimed at horizontally distributed text, and more attention should be paid to text positioning with morphological transformation in the future. Secondly, most of the datas used in this paper are industrial labels, and the detection effect of other industry labels still needs further experiments. Finally, the aspects of how to make the model more lightweight, faster in computation, and less expensive in experiments while ensuring accuracy are also aspects that we should further focus on subsequently.

REFERENCES

[1] D. Y. Wang, J. Su, and H. B. Yu, "Feature extraction and analysis of natural language processing for deep learning English language," *IEEE Access*, vol. 8, pp46335-46345, 2020.

[2] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR)," *IEEE Access*, vol. 8, pp142642-142668, 2020.

[3] S. Susan, and K. M. Rachna Devi, "Text area segmentation from document images by novel adaptive thresholding and template matching using texture cues," *Pattern Analysis and Applications*, vol. 23, no. 2, pp869-881, 2020.

[4] L. Neumann, and J. Matas, "A method for text localization and recognition in real-world images," *Proceedings of the Asian Conference on Computer Vision*, vol. 6494, pp770-783, 2010.

[5] C. G. Yan, H. T. Xie, J. J. Chen, Z. J. Zha, X. H. Hao, *et al*., "A fast Uyghur text detector for complex background images," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp3389-3398, 2018.

[6] S. Asgari Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad, and G. Hamarneh, "Deep semantic segmentation of natural and medical images: a review," *Artificial Intelligence Review*, vol. 54, no. 1, pp137-178, 2021.

[7] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. D. Wu, "Object detection with

deep learning: a review", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp3212-3232, 2019.

[8] D. Deng, H. F. Liu, X. L. Li, and D. Cai, "Pixellink, detecting scene text via instance segmentation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp2374-2382, 2018.

[9] E. Xie, Y. H. Zang, S. Shao, G. Yu, C. Yao, *et al*., "Scene text detection with supervised pyramid context network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp9038-9045, 2019.

[10] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp9365-9374, 2019.

[11] M. H. Liao, B. G. Shi, and X. Bai, "Textboxes++: a single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no.8, pp3676-3690, 2018.

[12] N. Anithadevi, J. Abinisha, V. Akalya, and V. Haripriya, "An improved SSD object detection slgorithm for safe social distancing and face mask detection in public areas through intelligent video analytics," *Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp1-7, 2021, doi: 10.1109/ICCCNT51525.2021.9579761.

[13] L. C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, *et al*., "Masklab: instance segmentation by refining object detection with semantic and direction features," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp4013-4022, 2018.

[14] R. Elanwar, W. Qin, M. Betke, and D. Wijaya, "Extracting text from scanned Arabic books: a large-scale benchmark dataset and a fine-tuned Faster-R-CNN model," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 24, no. 4, pp349-362, 2021.

[15] L. C. Cao, H. W. Li, R. B. Xie, and J. R. Zhu, "A text detection algorithm for image of student exercises based on CTPN and enhanced YOLOv3," *IEEE Access*, vol. 8, pp176924-176934, 2020.

[16] J. Redmon, and A. Farhadi, "Yolov3: an incremental improvement," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, doi: 10.48550/arXiv.1804.02767.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp779-788, 2016.

[18] J. Redmon, and A. Farhadi, "YOLO9000: better, faster, stronger," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp7263-7271, 2017.

[19] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, doi: 10.48550/arXiv.2004.10934.

[20] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, *et al*., "CSPNet: a new backbone that can enhance learning capability of CNN," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp390-391, 2020.

[21] Y. Y. Hu, X. X. Zhang, J. Yang, and S. Fu, "A hybrid convolutional neural network model based on different evolution for medical image classification," *Engineering Letters*, vol. 30, no. 1, pp168-177 , 2022.

[22] Z. F. Hu, L. Wang, Y. Luo, Y. L. Xia, and H. Xiao, "Speech emotion recognition model based on attention CNN BI-GRU fusing visual information," *Engineering Letters*, vol. 30, no. 2, pp427-434 , 2022.

[23] D. Li, J. Hu, C. H. Wang, X. T. Li, Q. She, *et al*., "Involution: inverting the inherence of convolution for visual recognition," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp12321-12330, 2021.

[24] J. P. C. Chiu, and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp357-370, 2016.