IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 27, 2020, accepted September 10, 2020, date of publication September 18, 2020, date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025221

A Text Detection Algorithm for Image of Student Exercises Based on CTPN and Enhanced YOLOv3

LANGCAI CAO[1,2], HONGWEI LI[1], RONGBIAO XIE[1], AND JINRONG ZHU[1]
[1]Department of Automation, Xiamen University, Xiamen 361005, China
[2]Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision, Xiamen 361005, China

Corresponding author: Langcai Cao (langcai@xmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772442.

ABSTRACT Intelligent learning system (ILS) has become a popular learning tool for students. It can collect students' wrong questions in exercises and dig out their unskilled knowledge points so that it can recommend personalized exercises for students. Detecting text accurately from images of students' exercises is significant and essential in an ILS. However, a big challenge of text detection is that traditional text detection algorithms can not detect complete text lines in an exercise scene, and their detection box always splits between Chinese and mathematical symbols. In this article, we propose a deep-learning-based approach for text detection, which improves You Only Look Once version 3 (YOLOv3) by changing the regression object from a single character to a fixed-width text and applies a stitching strategy to construct text lines based on the relation matrix, which improves the accuracy by 9.8%. Experimental results on both RCTW Chinese text detection dataset and real exercise scenario show that our model can improve detection effectiveness. In addition, we compare our method with two state-of-the-art approaches in applications of exercise text detection, and discuss its capability and limitations. We have also provided a platform which has implemented the proposal for detecting text lines in students' daily homework or examination papers, which enhances user experience well.

INDEX TERMS Text detection, exercise image, YOLOv3, CTPN, OCR platform, stitching algorithm.

I. INTRODUCTION

With the development of artificial intelligence, students' learning mode has changed dramatically. A variety of online education products from kindergarten to twelfth grade (K12) have emerged, such as Ape tutoring, Baidu homework help, etc. These new-type education products usually provide functions like online learning, online practicing, and searching questions by photos. Among all of these functions, searching questions by photos has been widely used. It can help students find correct answers for wrong questions by searching online databases in real-time, thus it increases the efficiency of solving problems. However, this procedure is instantaneous, which means students can not save their wrong questions. And also, these products cannot recommend appropriate exercises, which could enhance students' learning performances, for students based on each student's wrong questions as well.

We propose an intelligent learning system (ILS) which could recognize images uploaded by users and convert them to text, and then the results are saved in user's online question sets. In order to provide users appropriate exercises, we also leverage online question sets to mine users' unskilled knowledge points and incorporate their personal information to construct user profiles. We then adopt a recommendation system to provide users with appropriate exercises so that they could complement their flaws more efficiently and henceforth achieve higher scores in examinations.

For the purpose of quickly and easily uploading and synchronizing questions on exercise books or exam papers, a naive idea is to pre-store the index information (such as the catalog of learning materials) in the system so that users can use the catalog to find questions and manually add them to online question sets. However, this method requires to maintain the catalog for each learning material, which brings huge labor costs. For users, the idea is not simple enough and attractive.

To solve the above problem, we first adopt optical character recognition (OCR) technique to extract texts from question images which are uploaded by users, after that, we match the corresponding texts with the question database in real time,

The associate editor coordinating the review of this manuscript and approving it for publication was Xin Luo.

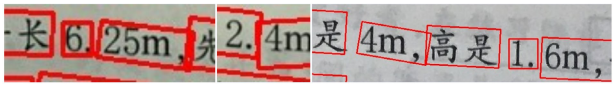


FIGURE 1. Images of text samples on an exercise image. The red box is usually broken at the junction of numbers and Chinese.

and add the matched results directly to each user's online question set, which is more convenient for users. The key elements of our project can be divided into the following 4 steps:

1) Text detection [1]–[5]: Text detection mainly analyzes the layout of the input image and locates the positions of texts in the image. Subsequently, it provides the text regions of the input image to the text recognition step;
2) Text recognition [6]–[8]: It converts text regions of the input image to machine readable strings;
3) Database matching: It matches the exercises in the database that are consistent with the recognized results;
4) Exercise recommendation: It utilizes question sets of users to mine unskilled knowledge points of each user and then recommends appropriate practice questions for each of them.

This article focuses on text detection in images and proposes a solution for problems encountered in text detection phase. Text detection is the basis of our whole project, only when the text detection stage finds locations of the texts in the image precisely, the recognizing and matching stages could output accurate results.

The biggest challenge in applying the text detection model to exercise images is that exercise images often contain many types of characters, such as number characters and Chinese characters, which increases the difficulty of text detection. As shown in Figure 1, different types of characters have different spacing. Besides, exercise images are often characterized by a large aspect ratio and texts are generally very long. Therefore, most existing text detection methods fail in detecting complete text lines in examination scenarios, and bounding boxes are usually broken at the junction of number text and Chinese text. Thus, one text line would get multiple detection boxes, and those boxes usually overlap or omit some characters, which has a bad effect on the subsequent text recognition stage. In our work, a way to detect text regions and a splicing method to concatenate fractures are proposed to solve the aforementioned problem, in which one has a role to detect text regions and the other one is responsible for splicing cracked bounding boxes into a complete bounding box.

In this article, an enhanced YOLOv3 model is presented to detect texts in primary school exercise images. In this scheme, a strategy based on detecting text regions rather than individual characters and a splicing algorithm are proposed to enhance the performance of text detection under examination paper scenarios. Specifically, we first fix the width of YOLOv3's anchor boxes and make it smaller, then we perform score threshold filtering and non-maximal suppression filtering on the output bounding boxes. Besides, we construct a relationship matrix to splice text lines instead of detecting them line-by-line like traditional text detection methods. We compare the effectiveness of our proposed method with some latest state-of-the-art text detection models under different improvement strategies in exercise images and it improves the accuracy of subsequent database matching stage. The contributions of this work are summarized as:

- We improve the YOLOv3 algorithm by determining regression object to a fixed-width text region so as to increase the accuracy of subsequent text recognition and database matching processes.
- A splicing strategy is proposed which concatenates detected text areas based on relation matrix and it brings a significant improvement in exercises scenarios.
- An open platform is built for text detection and text recognition in images. The platform is based on our model and it could support different improvement strategies.

The rest of the article is organized as follows. Section III discusses related works and Section III reviews some basic principle of our proposed method. Section IV presents our novel model, which performs well in exercise scenarios. Experimental tests are depicted in Section V, and Section VI introduces our OCR open platform. Section VII gives a conclusion of this article and provides possible future works.

II. RELATED WORK

Various types of text detectors have been proposed in recent years, which can be broadly classified into two categories, i.e., traditional text detectors and deep learning-based text detectors. Traditional text detectors leverage artificial low-level features and prior knowledge to distinguish text and non-text parts in the scene image. However, these algorithms lack robustness to various fonts and degraded images. In order to mitigate such a problem, a lot of deep learning-based text detection researches have been done, and they could achieve high performance. The majority of popular deep learning-based text detection approaches are inspired by semantic segmentation [9], [10] and object detection [11]–[15]. E.g., YOLOv3 [16] and Faster RCNN [17] are commonly used object detection algorithms that are applied directly to text detection missions.

In the range of detecting texts, there are several works that deserve to be mentioned. Baek *et al.* [18] proposed PixelLink which realizes text detection through instance segmentation. It performs two pixel-wise predictions: text/non-text prediction and link prediction. By setting two different thresholds, the pixel positive set and the link positive set can be obtained, and then the pixel positives are connected according to the link positive to obtain the connected components(CCs) set. Each element in the CCs set represents a text instance. Wang *et al.* [19] proposed the Progressive Scale Expansion Network (PSENet) which expands the small-scale kernel to the final text line size based on Breadth-First-Search(BFS). These segmentation-based methods are applied for processing multi-oriented texts in real scene images. However, once

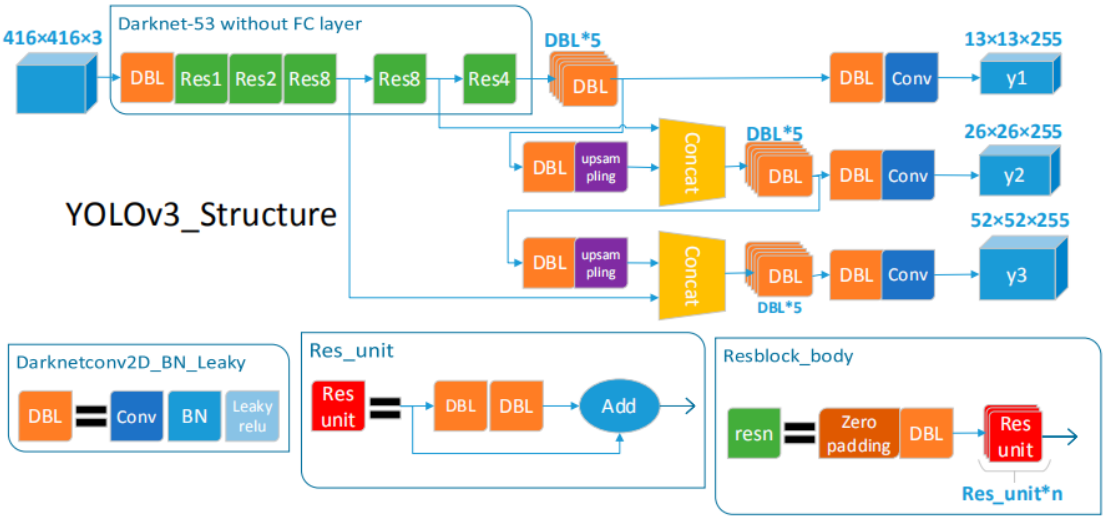


FIGURE 2. The YOLOv3 network structure.

the characters in the image are very close, separating them using only text/non-text semantic segmentation becomes extremely difficult.

Tian *et al.* [20] proposed CTPN that detects a small text boxes and judges whether they are text regions. When all small text boxes in an image are detected, those small text boxes belonging to the same text line are merged. After merging, all complete text boxes can be obtained. Although the idea of CTPN is novel and attractive, the time cost of CTPN is very expensive. Thereafter, Joseph Redmon *et al.* proposed YOLO series [16], [21], [22] that are of a new fully convolutional target detection as a regression problem, so as to quickly and accurately determine the position together with the type of the detected object. Baek *et al.* [18] proposed Character Region Awareness for Text Detection(CRAFT) which locates individual character regions and links detected characters to be a text instance. Liao *et al.* proposed TextBoxes++ [23] which is built on an end-to-end fully convolutional network and can detect texts in any direction. This type of target detection-based detectors need to design anchors or default boxes of various scales and different aspect ratios in advance.

Our text detector is also designed based on object detection. We set the detected target as a fixed-width text area, and integrate detected text areas using a splicing algorithm. Compared with text detection algorithm that uses a single character as the detected target, our method has higher accuracy and can cover the text more completely.

III. BASIC PRINCIPLE FOR TEXT DETECTION

A. YOLOv3

YOLOv3 [16] is an end-to-end object detection method, which directly regresses the position and category of the object in output [24], thus completes a speed-up detection. The network structure of YOLOv3 mainly includes two layers. The first layer is feature extraction that uses the Darknet-53 network to obtain a feature map, and sets the grid cells to the same size as the feature map. For each grid cell, three bounding boxes will be predicted. In addition, in the YOLOv3 network, the scale of feature maps used for object detection is divided into three ratios, namely 13×13, 26×26 and 52×52. The second layer is output process that produces location information (i.e.,$x, y, w, h, and\ pr$). Then, the unqualified bounding box will be removed by the filtering algorithm, and the text line construction algorithm is used to generate a text line, which is provided to the next stage for text recognition. YOLOv3 uses a construction method similar to the feature pyramid [25] in order to get three feature maps through two upsampling layers. It can detect objects with different sizes and meet the real-time detection. The specific network architecture of YOLOv3 is shown in FIGURE 2 [26].

B. CTPN

Tian *et al.* [20] proposed a connectionist text proposal network (CTPN) to localize text. In CTPN, a vertical anchor mechanism is developed to predict text locations in a fine scale [27], which greatly improves robustness and reliability on multi-scale and multi-language texts. It converts text detection into localizing fine scale text proposals, just detecting a part of a single character (i.e., image) to determine whether it is part of a character (such as the yellow box in FIGURE 3). After detecting all boxes, an in-network recurrent architecture is used to incorporate these small text proposals for getting a full-text box, then the text detection task is completed (such as the red box in FIGURE 3).

IV. OUR APPROACH

In this section, we introduce a text detection algorithm for student exercise images that includes two parts: one is the improved anchor box and the other one is the improved text line construction.

A. IMPROVED ANCHOR BOX

In our project, we found that most existing text detection methods fail in detecting complete text lines in exercise scenarios, and bounding boxes are usually broken at the junction

FIGURE 3. Sample diagram for CTPN detection method.

of number text and Chinese text. Thus, one text line would get multiple detection boxes, and those boxes usually overlap or omit some characters, which forces us to deal with missing text [14], [28] and repeated text. To detect a complete line of text, we give an enhanced version of YOLOv3 to solve the long text detection problem, mainly with the following innovations:

- We modified the aspect ratio of anchor boxes to make it suitable for detecting long texts.
- We turn the multi-class detection problem into a two-class of text/background detection problem.
- We utilize non-maximum suppression to improve the detection effect.

YOLOv3 is not good for detecting small objects and those who are close to each other. It detects objects by locating each character so that errors in the detection of each character will gradually be accumulated, which will lead to performance degradation. When aspect ratios of objects are unnormal, the generalization ability of YOLOv3 is weak. Therefore, we firstly improve this algorithm in terms of changing the detection object, that is, we only detect and determine whether it is a text area instead of locating characters.

In view of the fact that the text objects in the exercise images are mostly small targets, we try to improve the anchor box which has great effect on the object detection in YOLOv3. For predicting bounding boxes, we focus on changing the original three anchor boxes with different lengths and widths to three scale detections with fixed width but different heights, which can assign more accurate anchor boxes for text detection. This idea is inspired by CTPN, which hold the view that predicting the vertical position of text is easier than predicting the horizontal position of text.

The improved algorithm only recognizes the upper and the lower edges of the text instead of a single character. Hence, our algorithm only needs to detect the text area rather than the character. We in this paper take nine anchor boxes with size (8×11); (8×16); (8×23); (8×33); (8×48); (8×97); (8×139); (8×198); (8×283).

Anchor box is used in the convolutional neural network and each feature map in this network is divided into many cells, as shown in FIGURE 4. The convolutional neural network predicts four values for each bounding box on each cell, that is the offsets of the bounding box relative to the upper left

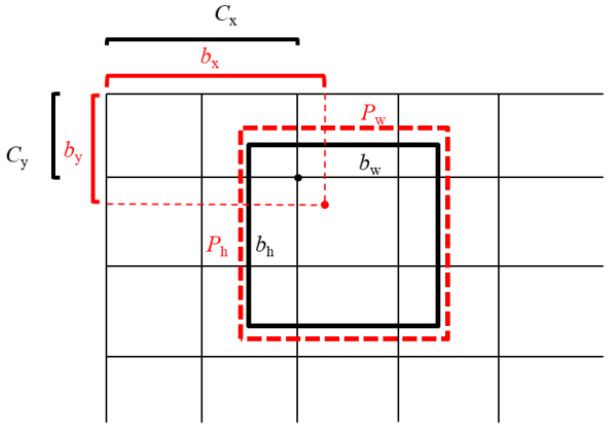


FIGURE 4. Bounding boxes with dimension priors and location prediction.

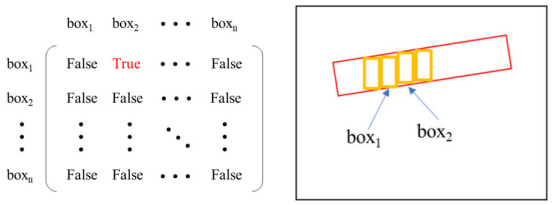
corner of the cell t_x, t_y and the scale relative to the anchor box t_w, t_h. Suppose that the cell is offset relative to the upper left corner of the image by (c_x, c_y), the size of the input layer and output layer is i_w, g_w, and the anchor box has height p_h, then the predicted bounding box's (i.e., target's) coordinates (b_x, b_y) and the bounding box's width b_w and height b_h compute

$$\begin{cases} b_x = \dfrac{i_w}{g_w} \times (\text{sigmoid}\,(t_x) + c_x) \\ b_y = \dfrac{i_w}{g_w} \times \left(\text{sigmoid}\left(t_y\right) + c_y\right) \\ b_w = 8 \times e^{t_w} \\ b_h = P_h \times e^{t_h}. \end{cases}$$

The localization results will be many boxes with partial overlap between them, thus we use Non-Maximum Suppression (NMS) [29] to select the bounding boxes with highest score, and ensure that multiple targets have localization results.We evaluate the probability that the bounding box contains texts. If the overlapping ratio of the bounding box and the actual target bounding box is greater than that of any other bounding box, the probability of this anchor box is 1. If the overlapping ratio is greater than 0.5, but not the maximum, this prediction is ignored. If the bounding box is not considered to contain an text area, it has no effect in the loss function.

Loss function is an important evaluation basis for parameter optimization and update of the deep network. For our algorithm, the loss function for text detection is calculated by combining the predicted value and the ground truth. The network predicts the coordinates of the bounding box, objectness prediction and class predictions. Usually, YOLOv3 is used for multi-object classification, i.e. the default box at each level will classify all classes of objects (plus background) for scoring. We modify it to binary classification problem of text/background to better fit the text recognition requirements, and the loss function is modified as follows:

$$loss_{class} = \sum_{r=0}^{k-1} \left((r == \text{truth}_{\text{class}})?1 : 0 - \text{predict}_{\text{class}_r}\right)^2$$



(a) Relation matrix (b) Adjacent box

FIGURE 5. Text line construction. The box_1 and box_2 are adjacent to each other in (b), and the corresponding value of (box_1,box_2) in (a) is True.

The total loss of the network is composed of four types of loss, the formula is as follows:

$$loss_{total} = loss_{xy} + loss_{wh} + loss_{confidence} + loss_{class}.$$

Among them, $loss_{xy}$, $loss_{confidence}$, and $loss_{class}$ are all in the range of 0 and 1, so we calculate them by using the binary cross-entropy loss function. While for $loss_{wh}$, the mean squared error (MSE) is used to calculate the loss function due to the wide range of its values.

B. IMPROVED TEXT LINE CONSTRUCTION

We propose a bounding box stitching strategy based on the relation matrix, which improves the robustness of the model in different scenarios. It has better adaptability to oblique text, and can get complete text lines. The text box will not broke due to the large gap between different types of characters. This algorithm contains two main parts: construction of the relation matrix of the bounding boxes and calculation of the text line.

The relation matrix of the bounding boxes is a two dimensional Boolean matrix, which describes the right adjacent box of each bounding box. As shown in FIGURE 5(b), box_1 and box_2 are adjacent in the red text line, and box_2 is the right adjacency of box_1, so the position (box_1, box_2) in the relation matrix is assigned True, as shown in FIGURE 5(a).

1) CONSTRUCT THE RELATION MATRIX OF THE BOUNDING BOXES

For each target bounding box (abscissa is x), we will find its right adjacency box and build a relationship set for it, which is a series of bounding boxes with abscissas in $[x, x+mhg]$ and meets the overlap-similar condition defined in the following. And mhg is an artificial horizontal threshold, whose value is usually 0.08 times the image's width. To ensure that the calculated right adjacency box and the target bounding box are in the same text line, the overlap-similar calculation is performed on the box between $x - mhg$ and x, thresholds are given artificially.

Take box_1 and box_2 as an example, the overlap and similar is calculated as

$$\begin{aligned} y_{min} &= max\,(y_{10}, y_{20}), \\ y_{max} &= min\,(y_{13}, y_{23}), \\ \text{overlap} &= \frac{y_{max} - y_{min} + 1}{min\,(h_1, h_2)}, \\ \text{similar} &= \frac{min\,(h_1, h_2)}{max\,(h_1, h_2)}, \end{aligned}$$

Algorithm 1 Subgraph

Require: graph = $[[g_{11}, g_{12}, \cdots, g_{1n}], \cdots, [g_{n1}, g_{n2}, \cdots, g_{nn}]]$;
Ensure: sub_graphs = $[line_1, line_2, \cdots, line_m]$;

```
1:  Initialize sub_graphs = ∅;
2:  for index in graph.shape[0] do
3:    if not graph[:, index] and graph[index,:] then
4:      k←index, line ← [k];
5:      while graph[k,:] has True: do
6:        k←graph[k,:].whereTrue();
7:        Update line ← line ∩ {k};
8:      end while
9:    end if
10:   Update sub_graphs ← sub_graphs ∩ {line};
11: end for
12: return sub_graphs;
```

where y_{10} and y_{13} are the upper left and the lower left vertical coordinates of box_1 respectively, h_1 is the height of box_1, y_{20} and y_{23} are the upper left and the lower left vertical coordinates of box_2 respectively, and h_2 is the height of box_2.

We take the box that closest to the target bounding box and meets the overlap-similar condition in the relationship set as the right adjacency box. After the relation matrix is established, we determine text lines according to it by dividing the bounding boxes belonging to the same text line into a set. Then text lines of a picture creates.

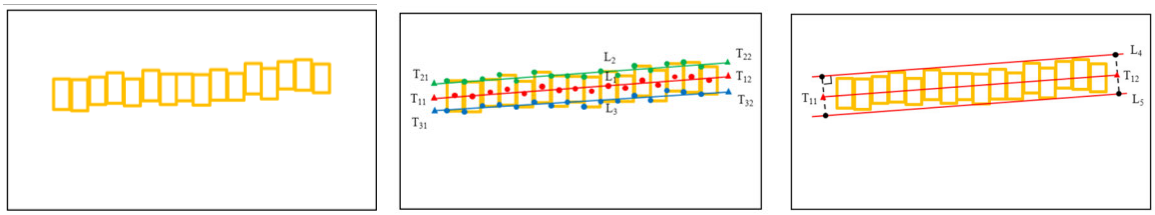
We simply describe this process in Algorithm 1. The general idea is to find the bounding box that is the first box of each text line (line 2), then we find its right adjacency box in the matrix (lines 3-9), and iterate this process until all bounding boxes are divided into a certain set (line 10).

2) TEXT LINE CALCULATION

We calculate the coordinate information of text lines according to the bounding box set of each text line, as depicted in FIGURE 6(a) that shows a set of bounding boxes belonging to a certain text line. Let $lb_i = [box_1, box_2, \ldots, box_n]$ be the bounding box set of text line i, where $box_n = [x_{left}, y_{upper}, x_{right}, y_{lower}]$. In the bounding box, x_{left} is the abscissa of the lower (upper) left point, y_{uper} is the vertical coordinate of the upper left point, x_{right} is the abscissa of the lower (upper) right point and y_{lower} is the vertical coordinate of the lower left point. We then simply describe the calculation of text line in Algorithm 2.

First, we extract the center point c, upper left point and lower left point of each bounding box (lines 1-3), and use the least square method (i.e., *least_square_fit*) to fit three straight lines L_1, L_2 and L_3 (lines 4-6), which are used as the baseline for the final text box extraction, as shown in FIGURE 6(b). And the slope and intercept of L_1, L_2 and L_3 are

$$\begin{aligned} k_i &= \bar{y}_i - b_i\bar{x}_i, \quad i = 1, 2, 3, \\ b_i &= \frac{\sum_{j=0}^{n} x_j y_j - \bar{y}_i\bar{x}_i}{\sum_{j=0}^{n} x_j^2 - n\bar{x}_i^2}, \quad i = 1, 2, 3, \end{aligned}$$



(a) BOX set (b) Least squares line fitting (c) Text line coordinate calculation

FIGURE 6. Box coordinate straight line fitting.

(a) Original label (b) Label divided by character (c) Label divided by fixed width

FIGURE 7. Different ways to divide labels. Labels divided by characters in (b) are used to train YOLOv3, and labels divided by fixed width in (c) are used to train our model.

where $\bar{y}_i$ is the average vertical coordinate of the center points, upper left points or lower left points and $\bar{x}_i$ is the average abscissas of the center points, upper left points or lower left points.

Second, we get the minimum value x_{min} and the maximum value x_{max} of the abscissa in the bounding box set (lines 7-8) and the width b_w of bounding box, where b_w is the width of anchor box. The $x_{min} - b_w$ and $x_{max} + b_w$ points are brought into lines L_1, L_2, and L_3, and then six coordinate points $T_{11}(x_{min} - b_w, y_{l1})$, $T_{12}(x_{max} + b_w, y_{r1})$, $T_{21}(x_{min} - b_w, y_{l2})$, $T_{22}(x_{max} + b_w, y_{r2})$, $T_{31}(x_{min} - b_w, y_{l3})$, and $T_{32}(x_{max} + b_w, y_{r3})$ are obtained (lines 9-11), as shown in the six small triangles of FIGURE 6(b).

Next, we calculate the distance Δ_{top} from points T_{21} and T_{22} to the line L_1 and the distance Δ_{down} from points T_{31} and T_{32} to the line L_1, where d $(\cdot)$ denotes the distance from a point to a line. Assuming the straight-line equation of L_1 is $L_1 = k_1 x + b_1$, we solve the parallel lines L_4 and L_5 as

$$\begin{cases} \mathrm{L}_4 = k_1 x + \Delta_{top} \times \sqrt{k_1^2 + 1} + b_1, \\ \mathrm{L}_5 = k_1 x - \Delta_{down} \times \sqrt{k_1^2 + 1} + b_1, \end{cases}$$

where Δ_{top} and Δ_{down} are also represented as vertical distances from L_4 and L_5 to the line L_1 respectively (lines 12-15).

Finally, the footpoints of T_{11} to L_4 and L_5 and the footpoints of T_{12} to L_4 and L_5 are the four coordinates of the text box (line 16), as shown in FIGURE 6(c). In line 16, the footpoints are denoted by fp$(\cdot)$, such as fp$(T_{11}$ to $L_5)$ denotes the footpoints' coordinates of T_{11} to L_5. Thus, we can use these four footpoints to get the position of the i_{th} text line, which are denoted as tc_i in Algorithm 2.

Algorithm 2 Generating Text Line

Require: bounding box set of the i_{th} text line lb_i, the width of anchor box b_w, the slope k_i and the intercept b_i;

Ensure: the ordinates of footpoints to the i_{th} text line tc_i;

1: Initialize $tc_i = \emptyset$;
2: $x_c \leftarrow (lb_i[:, 0] + lb_i[:, 2])/2$;
3: $y_c \leftarrow (lb_i[:, 1] + lb_i[:, 3])/2$;
4: $L_1 \leftarrow least\_square\_fit(x_c, y_c)$;
5: $L_2 \leftarrow least\_square\_fit(lb_i[:, 0], lb_i[:, 1])$;
6: $L_3 \leftarrow least\_square\_fit(lb_i[:, 0], lb_i[:, 3])$;
7: $x_{min} \leftarrow \min(lb_i[:, 0])$;
8: $x_{max} \leftarrow \max(lb_i[:, 0])$;
9: $y_{l1} \leftarrow k_1(x_{min} - b_w) + b_1$, $y_{r1} \leftarrow k_1(x_{max} + b_w) + b_1$;
10: $y_{l2} \leftarrow k_2(x_{min} - b_w) + b_1$, $y_{r2} \leftarrow k_2(x_{max} + b_w) + b_1$;
11: $y_{l3} \leftarrow k_3(x_{min} - b_w) + b_1$, $y_{r3} \leftarrow k_3(x_{max} + b_w) + b_1$;
12: $\Delta_{top} \leftarrow \max(\mathrm{d}(T_{21}$ to $L_1), \mathrm{d}(T_{22}$ to $L_1))$;
13: $\Delta_{down} \leftarrow \max(\mathrm{d}(T_{31}$ to $L_1), \mathrm{d}(T_{32}$ to $L_1))$;
14: $L_4 \leftarrow k_1 x + \Delta_{top} \times \sqrt{k_1{}^2 + 1} + b_1$;
15: $L_5 \leftarrow k_1 x + \Delta_{down} \times \sqrt{k_1{}^2 + 1} + b_1$;
16: Update $tc_i \leftarrow tc_i \cup (\mathrm{fp}(T_{11}$ to $L_4)$, $\mathrm{fp}(T_{11}$ to $L_5)$, $\mathrm{fp}(T_{12}$ to $L_4)$, $\mathrm{fp}(T_{12}$ to $L_5))$;
17: **return** tc_i;

V. EXPERIMENTAL RESULTS

We first study the impact of different setting of improvement strategies on detection performance and then evaluate our proposed model against two state-of-the-art approaches. These experiments are implemented on a Linux workstation, with an Intel Core CPU 3.40GHz 16GB RAM through Python programming.

A. DATASETS AND PREPROCESSING

We choose RCTW Chinese text detection dataset [20] of the ICDAR 2017 competition and a real exercise dataset of primary school exercise book. The RCTW-17 contains a large-scale dataset that consists of various kinds of images, including street views, posters, menus, indoor scenes, and screenshots. The real exercise dataset is obtained by image capturing devices (i.e., scaners), which has two sub-

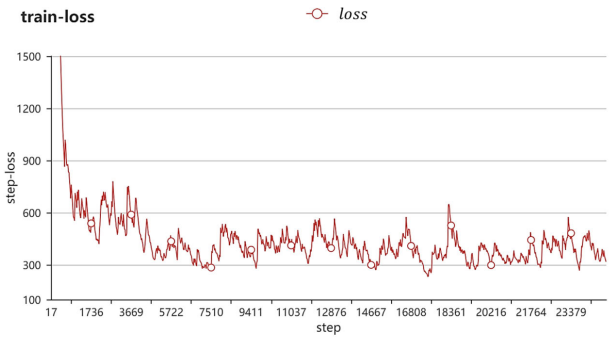


FIGURE 8. The loss curves of our method for training set.

sets ('math','Chinese') classified by subject. We labeled it with LabelMe. There are totally 8033 textual instances, 7229 of which are used for training and remaining are for testing.

The label contains coordinates and character information, and the characters that are difficult to recognize are indicated by '#', as shown in FIGURE 7(a). Due to the difference between YOLOv3 and our proposed method, the labeled text lines of the dataset are equally segmented. The images for training YOLOv3 are labeled with single-character text boxes, as shown in Figure 7(b), and the images for training our model are labeled with fixed-width text regions, as shown in Figure 7(c).

B. IMPLEMENTATION DETAILS

We trained the network by setting the epoch to 5. Each epoch contains 7229 steps. The learning rate is set to 0.005 and decay is set to 0.3. Overlap and Similar are applied for determine adjacent bounding boxes and the threshold is 0.7. All comparison methods in text detection experiments are pre-trained on ICDAR 2017 dataset and then fine-tuned on exercise dataset.

The loss curve of the train set is shown in FIGURE 8. It can be found that the loss decreases rapidly within 2000 steps, and then the decline gradually decreases. The loss during the entire training process decreases from 31025.2422 to about 253.6912, which indicates that the training process is effective.

TABLE 1. Performance on the Exercise Dataset.

Model	Search Accuracy	Search Time (Average)	Time (Text Box Detection)
YOLOv3	79.80%	4.18s	0.86s
IMP-Obj	86.70%	5.12s	1.65s
IMP	89.60%	4.71s	1.58s

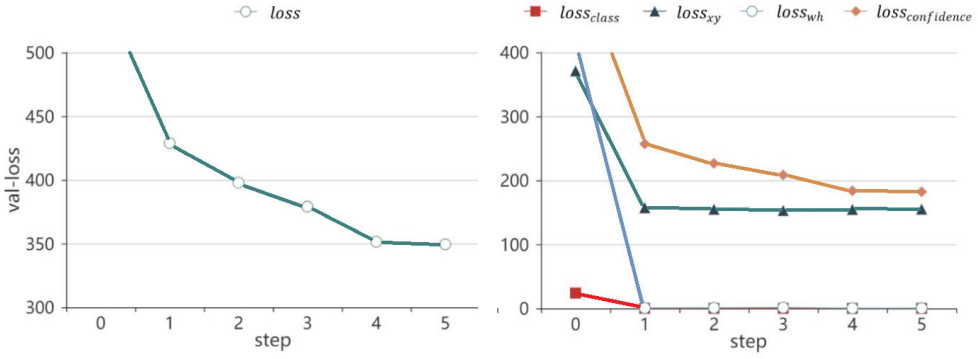
We evaluate the trained model on the test set. FIGURE 9(a) shows the trend of total loss on each epoch, and FIGURE 9(b) shows the trend of four sub-losses in detail. After the first epoch, the $loss_{wh}$ and $loss_{class}$ have approached zero, and the $loss_{confidence}$ continues to converge as the training progresses and eventually remains at 350.

C. EVALUATION ON THE EXERCISE IMAGES

We focus on the impact of different strategies on detection effect. Three models are tested on one exercise dataset, namely the original YOLOv3 model, the model that only improves the detection object, and the model that combines all strategies.

[Metrics] Since the ultimate goal of the method implemented in this paper is to get the text of the exercise images uploaded by users in combination with text recognition algorithm, and then perform a database matching. Here the step of exercise matching searches the database for corresponding exercises based on the detected and recognized results. We therefore evaluate the performance with **Search Accuracy**, **Search Time** and **Time(Text Box Detection)**. Search accuracy reflects how many correct texts were searched correctly. Since text recognition is not the focus of this article, a basic model Densenet and CTC is adopted to recognize the texts.

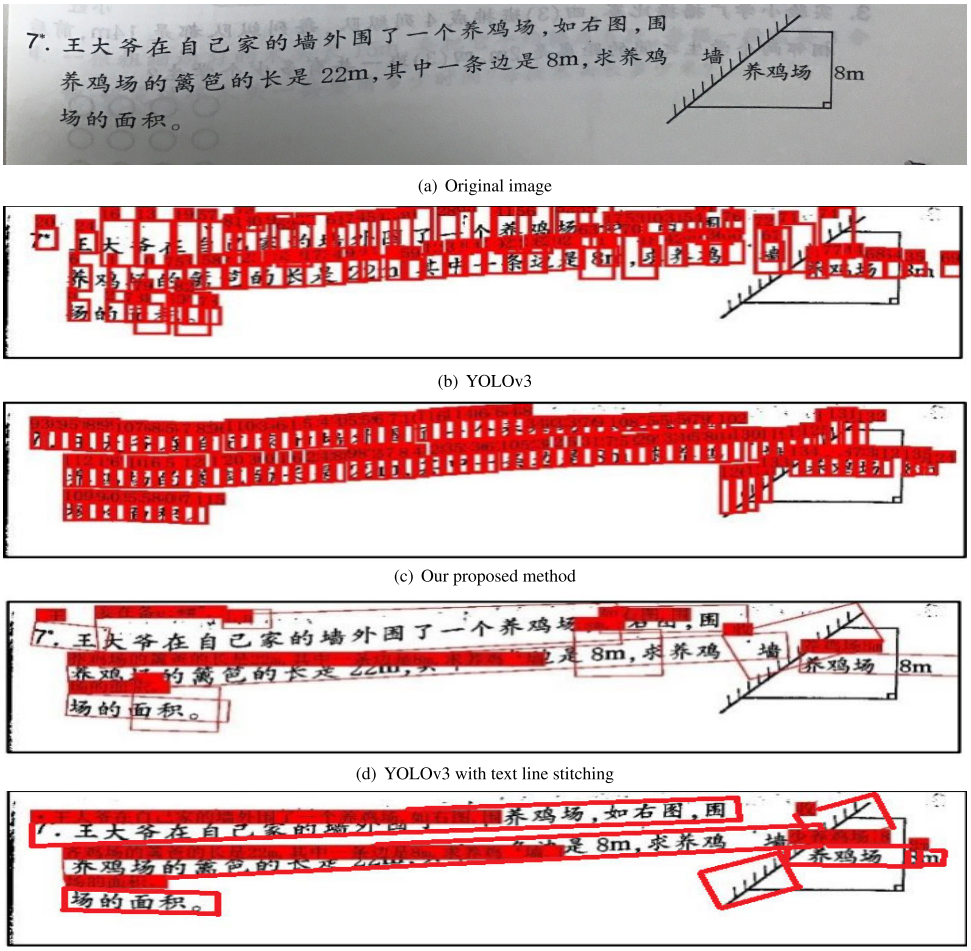
[Results] The results are summarized in TABLE 1. It shows: (1)YOLOv3, basic model; (2)IMP-Obj, the model that only improves the detection object; (3)IMP, the model that combines all strategies. The experimental results show that the Search Accuracy of the IMP model increased by 9.8% compared with YOLOv3 model, and improvement of the detection object algorithm(IMP-Obj) also had significant improvement on the Search Accuracy. As the input layer of YOLOv3 is increased from 416 × 416 to 608 × 608,



(a) Total loss on test set

(b) Sub-loss on test set

FIGURE 9. Loss curves of model testing.



(a) Original image

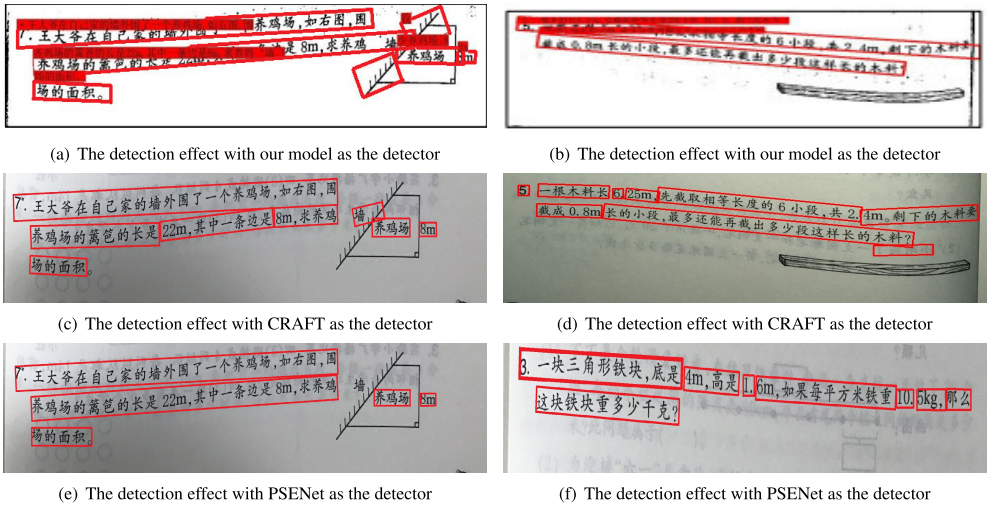
(b) YOLOv3

(c) Our proposed method

(d) YOLOv3 with text line stitching

(e) Our method with text line stitching

FIGURE 10. **Text detection results via YOLOv3 and the proposed improved method as well as comparative results using text line stitching.**



(a) The detection effect with our model as the detector

(b) The detection effect with our model as the detector

(c) The detection effect with CRAFT as the detector

(d) The detection effect with CRAFT as the detector

(e) The detection effect with PSENet as the detector

(f) The detection effect with PSENet as the detector

FIGURE 11. **Comparison of text detection effects between two latest model and our model.**

the detection time of the two new models naturally increase from 0.86s to 1.65s and 1.58s, respectively. Although the improved model is not the fastest, it is acceptable in our project. IMP-Obj performs better than the original YOLOv3 because it uses improved anchor boxes and loss function to obtain a detection box with higher

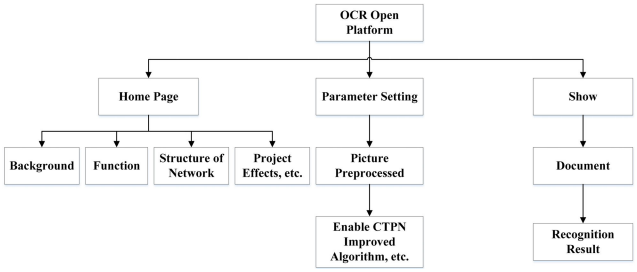


FIGURE 12. OCR platform framework.

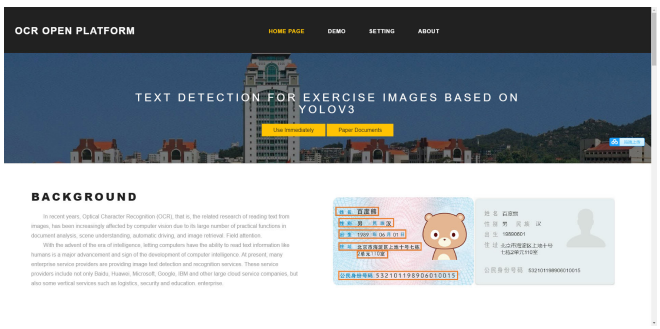
text confidence. IMP had higher performance than the original YOLOv3 because it improved the method of generating text lines, which used the indicators of height and distance to comprehensively determine the adjacency matrix, and utilize adjacency matrix as the basis for merging text lines. YOLOv3 adopts a line-by-line detection method to cluster the detection boxes into a collection. It is not suitable for text with multiple fonts and a large aspect ratio.

FIGURE 10(b)(d) show that the text box (i.e., bounding box) detected by YOLOv3 is relatively irregular and has large overlap which will cause a great obstacle to the subsequent text splicing algorithm. Our proposed model can yield regular-sized detection boxes with minimal overlap among detection boxes, making the spliced text lines to cover the text area completely, as shown in FIGURE 10(c)(e).

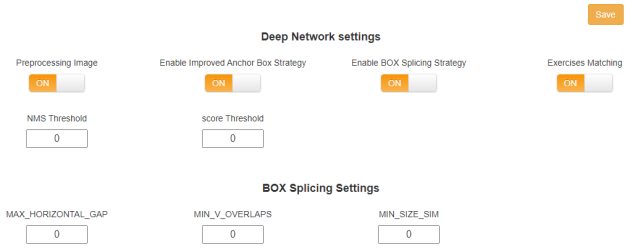
D. COMPARISON WITH STATE-OF-THE-ARTS

At last, we also compare our model with Character Region Awareness for Text Detection (CRAFT) [18] and Shape Robust Text Detection with Progressive Scale Expansion Network(PSENet) [19]. CRAFT is a text detection algorithm with good robustness to scale changes and long text detection. PSENet uses a progressive expansion algorithm to expand the small-scale kernel to the final text line. The comparison in this section focuses on the detection performance of the above two state-of-art methods and our model. Either CARFT, PSENet or our method as a detector, horizontal Chinese character text detection yields good results. However, the exercise text usually includes numbers and mathematical symbols, and the inconsistent interval between numbers and Chinese characters makes difference in detection performance.

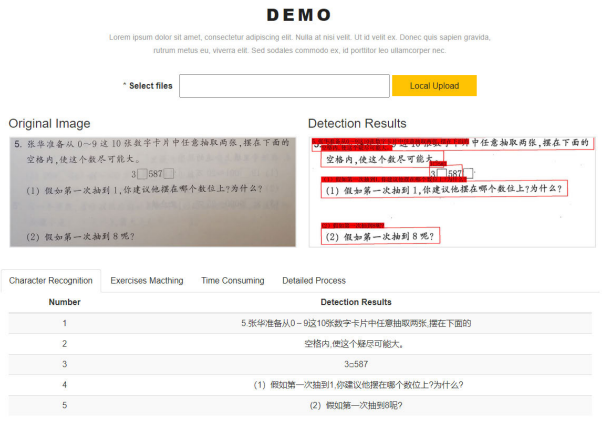
In the case of the text contains both Chinese and other characters such as numbers, different algorithms act as detectors to detect the text, as shown in Figure 11. It can be shown that CRAFT's detection box separated and failed in detecting a complete text box at the junction of numbers characters and Chinese characters, whereas our method is successful in detecting a complete line of text. This is because CRAFT determines the final text line based on the connection relationship among the detected characters. When the interval among characters is uneven or large, the algorithm will judge



(a) The home page



(b) The setting page



(c) The demo page

FIGURE 13. OCR platform framework.

that it has no connection relationship and cannot obtain a complete text box. The results of PSENet are also very good, but there are still some text lines that cannot be fully detected. PSENet predicts the different kernel sizes of the text line, and then uses a progressive expansion algorithm to expand the small-scale kernel to the final size of text line. Due to the fact that there is a relatively large margin between the small-scale kernels, it can distinguish adjacent text lines well. But at the same time, it also divide the one text line with large intervals into multiple boxes.

VI. OCR PLATFORM

Our platform uses optical character recognition to directly convert the text of images into editable text. It can recognize the content of students' daily homework and examination papers, the automatic entry of student homework and examination papers, and improve the efficiency of teachers' teaching and students' learning.

It also can choose different parameters to find a more suitable configuration and showing details of the models. FIGURE 12 is a functional framework of the OCR platform including the home page, the setting page, and the demo page.

As shown in FIGURE 13(a), the home page shows details of models in this paper including the network architecture, the performance of the project, the summary outlook of future real-time OCR and the current experimental results, which makes the structure of model clearer.

FIGURE 13(b) is the setting page. We can intuitively set the structural parameters of our models on this page, i.e., whether to perform image preprocessing, whether to improve the object detection, whether to enable box stitching and set thresholds. And multiple models can be trained on different settings.

After operating on the setting page, we can run the relative model in the demo page, shown in FIGURE 13(c). We first upload a picture and then test the model to get the result. Of course, this page can be used to understand, debug and demonstrate the model.

VII. CONCLUSION

This paper focuses on text detection based on the exercise scenarios and proposes a text detection algorithm in terms of improving the advanced real-time object detection network YOLOv3 for enhancing the accuracy of text detection. This new algorithm includes two main parts. One is changing the detection objects by improving the anchor box. The other one is a bounding box splicing algorithm based on the relation matrix. It makes the detection frame of text detection more regular, the overlap area smaller, covers the text area more fully, and is more robust to oblique text. Compared with the original YOLOv3 algorithm, our algorithm improves the accuracy by 9.8%, and the running time of the entire model is 1.58s. The experimental part uses ICDAR2017's RCTW Chinese dataset and manually labeled the scene problem dataset to perform weight optimization. Through a series of scoring threshold filtering, NMS filtering, and text line construction algorithms, the precise positioning of text lines is obtained and provided to subsequent sequential text recognition algorithms.

Although the existing algorithms have successfully applied in most application scenarios, we still have some follow-up works. First, our algorithm recognizes the background as text in some scenarios to a certain extent, and it mainly targets for objects or horizontally distributed text. The algorithm needs further research on the text detection effect of some curved or circular distribution cases. In the end, although the current detection speed of our algorithm has reached the industrial level, it is still a big challenge for real-time text detection to be used in scenarios such as autonomous driving, and even real-time OCR recognition. It needs to be improved on hardware, calculation speed and model compression.

REFERENCES

[1] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2963–2970.

[2] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, Sep. 2004.

[3] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4159–4167.

[4] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 256–268, Apr. 2002.

[5] P. P. Roy, U. Pal, J. Lladós, and M. Delalandre, "Multi-oriented and multi-sized touching character segmentation using dynamic programming," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 11–15.

[6] Y. K. Ham, S. K. Min, K. C. Hong, R. H. Park, and G. T. Park, "Recognition of raised characters for automatic classification of rubber tires," *Opt. Eng.*, vol. 34, no. 1, pp. 102–109, 1995.

[7] R. Smith, "An overview of the tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 2, Sep. 2007, pp. 629–633.

[8] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3538–3545.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

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

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, no. 2, pp. 1097–1105, 2012.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision—ECCV*, vol. 9905. Cham, Switzerland: Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[14] X. Luo, M. Zhou, S. Li, and M. Shang, "An inherently nonnegative latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2011–2022, May 2018.

[15] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767
[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[18] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9357–9366.
[19] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9328–9337.
[20] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 56–72.
[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[23] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.
[24] Z. Deng, R. Yang, R. Lan, Z. Liu, and X. Luo, "SE-IYOLOV3: An accurate small scale face detector for outdoor security," *Mathematics*, vol. 8, no. 1, p. 93, Jan. 2020.
[25] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[26] Y. Zhou, C. Wu, Q. Wu, Z. M. Eli, N. Xiong, and S. Zhang, "Design and analysis of refined inspection of field conditions of oilfield pumping wells based on rotorcraft UAV technology," *Electronics*, vol. 8, no. 12, p. 1504, Dec. 2019.
[27] H. Lin, P. Yang, and F. Zhang, "Review of scene text detection and recognition," *Arch. Comput. Methods Eng.*, vol. 27, no. 2, pp. 433–454, Apr. 2020.
[28] X. Luo, M. Zhou, S. Li, L. Hu, and M. Shang, "Non-negativity constrained missing data estimation for high-dimensional and sparse matrices from industrial applications," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1844–1855, May 2020.
[29] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6469–6477.

LANGCAI CAO received the Ph.D. degree in automation from Xiamen University, in 2011. He is currently an Associate Professor with Xiamen University. His research interests include research and development of information systems, process intelligence, and machine learning.

HONGWEI LI received the bachelor's degree in automation from Northeast Petroleum University, in 2019. Her research interests include optical character recognition and social networks.

RONGBIAO XIE received the bachelor's degree in automation from Xiamen University, in 2019. His research interests include optical character recognition and reinforcement learning.

JINRONG ZHU received the master's degree in automation from Xiamen University, in 2020. Her research interests include optical character recognition and community search.