Signal, Image and Video Processing
https://doi.org/10.1007/s11760-019-01525-1

ORIGINAL PAPER

Detonator coded character spotting based on convolutional neural networks

Guandong Cen[1] · Nian Cai[1] · Jixiu Wu[1] · Feiyang Li[1] · Han Wang[2] · Guotian Wang[3]

Received: 22 November 2018 / Revised: 2 May 2019 / Accepted: 15 June 2019

Abstract
To facilitate the management of detonators, an automatic spotting system is proposed for detonator coded characters based on convolutional neural networks. The system contains a multi-scale detection network, a multi-label recognition network and a post-processing layer. An improved fully convolution network (FCN) is designed as the multi-scale detection network to obtain the accurate response map of the detonator image. Two subnetworks are parallel integrated into the FCN to perform a coarse-to-fine detection. An improved Jaccard loss function with a regularization term is defined to train the FCN. The region of interest (ROI) of the detonator image is achieved when the response map is post-processed by the post-processing layer. Finally, a modified multi-label network is used to recognize the detonator coded characters in the ROI. The experimental results indicate that the proposed system achieves a better spotting performance for detonator coded characters than the state-of-the-art text spotting methods in terms of accuracy and efficiency.

Keywords Detonator coded character · Text spotting · Convolutional neural network · Fully convolutional network · Jaccard loss function

1 Introduction

Detonator is a type of explosive materials, which is often used in engineering blasting. To facilitate the management, a unique serial number is carved in each detonator by using laser engraving technology. Nowadays, these detonators are managed by manually checking the serial numbers of detonators. However, this task is time-consuming, boring and fallible. Therefore, it is necessary to develop an automatic spotting system for detonator coded characters to manage the detonators efficiently.

There are only a few researches on detonator coded character recognition [1, 2]. In [1, 2], the authors directly used CCD (charge-coupled device) cameras to capture detonator images involving coded characters. It means that detonators should be manually rotated so that the cameras can aim at the regions where the coded characters are carved. If a machine is used to automatically rotate the detonators, the mechanical structure of the image acquisition system may be complicated. Also, some distortions exist in the detonator coded characters since the detonator is a cylinder, which will result in misrecognition. In addition, their acquisition systems will meet with the safety problem since the detonators are exposed to the environment. So, how to safely acquire high-quality detonator images at a low cost is a challenge for automatically recognizing detonator coded characters.

The acquired detonator images are possibly blurred, of low contrast and with a lot of noise if the detonators are rusty. It means that there is no distinct boundary between the foreground (the coded character region) and the background of the detonator image. The traditional image processing methods [1, 2] will meet with the difficulties although they have been employed for detonator coded character recognition. In [1], a series of traditional image processing methods are employed, which include deflection correction, area segmentation, binarization, character segmentation and support vector machine (SVM). Here, the region of interest (ROI) of the detonator image is detected by area segmentation based on Hough transformation, which is a vital step for recog-

✉ Nian Cai
cainian@gdut.edu.cn

1 School of Information Engineering, Guangdong University of Technology, Guangzhou, China

2 School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China

3 Guandong Face Intelligent Technology Co. Ltd, Guangzhou, China

Published online: 21 June 2019

nition. However, Hough transformation is not a promising segmentation method when it is used for the blurred and low-contrast images with a lot of noise. Zhu et al. [2] proposed a robust character segmentation for detonator coded character recognition, which was based on adaptive stroke width estimation and simple linear iterative clustering super-pixel (SLIC super-pixel) region growing. However, they focused on individual character segmentation for the detonators, which assumed that the ROI of the detonator image involving coded characters was well detected. In practice, ROI detection is an important step of text recognition before individual character segmentation.

Although some researchers proposed several solutions to domain-specific text detection [3–5], they should elaborately extract the features from the images. Nowadays, deep learning is a promising approach to detecting the text ROIs of natural images because of its excellent ability to automatically extract self-learned features [6–11]. An improved single-shot multi-box detector (SSD) was proposed for text detection in [9], which incorporated long stripe default boxes and long stripe convolution kernels into the traditional SSD [12]. Connectionist text proposal network (CTPN) combines a convolutional neural network (CNN) and a bidirectional long short-term memory (Bi-LSTM) network to detect the texts [10]. A 3×3 window is slid through feature maps of the last layer of the CNN, and a convolutional feature is achieved for each window. Subsequently, the convolutional features are fed into a Bi-LSTM network to predict texts and non-texts. A fully convolutional network (FCN) was introduced into text detection [11], considering it as a text/non-text semantic segmentation problem. In [11], a text block FCN and a centroid FCN are combined to achieve the response maps of texts. Then, the multi-oriented texts are detected after a post-processing procedure. Compared with detonator images, natural images have complicated scenes and relatively high quality. So, these deep learning methods design and optimize their networks to deal with the problems how to detect the texts from complicated and high-quality scenes. They may not excellently detect the ROIs of detonator images since detonator images often have low contrast. Furthermore, these methods have complicated network architectures to accurately detect the texts, which indicate that the detection speeds will be low if they are implemented in CPU. To increase the detection speeds, they are often implemented in GPU. However, the GPU hardware is too expensive and rarely employed in real industries.

To deal with the above problems, an image acquisition system is designed to safely acquire detonator images. Also, an automatic spotting system, which is based on an improved FCN and a modified multi-label CNN, is designed to accurately and time-efficiently detect and recognize detonator coded characters. The image acquisition system includes an explosion proof box, a linear CCD camera, two white LED light sources and a rotating motor. The automatic spotting system involves a multi-scale detection network, a post-processing layer and a multi-label recognition network. The multi-scale detection network is designed to achieve the response map of the detonator image, which includes two subnetworks. Compared with most of the existing CNNs, each subnetwork has few layers and few feature maps in each layer. Thus, these can accelerate the detection, which indicates that the proposed spotting system can be quickly implemented in CPU. Different from traditional FCNs with the weighted sigmoid cross-entropy loss function (WSCEL), the training strategy of our detection network is based on Jaccard loss function (JLF) [13], which is helpful to generate the more accurate response map. However, some undesired responses to the background around the ground-truth text may be obtained by the detection network with the original JLF, greatly influencing the performance of text detection. Therefore, an improved Jaccard loss function (IJLF) is defined to suppress these undesired responses, which involves a regularization term compared with the original JLF. Then, a post-processing layer followed by the detection network is embedded to effectively integrate the detection and recognition networks. Next, the ROI obtained by the post-processing layer is input into a modified multi-label CNN. Finally, the detonator coded characters in the image are directly recognized by the modified multi-label CNN.

The rest of this paper is organized as follows. Section 2 describes the image acquisition system. Section 3 introduces the proposed spotting system including the detection network with the IJLF, the recognition network and the post-processing layer. Section 4 shows the experimental results and discussions. Finally, a conclusion is drawn in Sect. 5.

2 Image acquisition system and the dataset

Detonator images are collected by an image acquisition system shown in Fig. 1. The system includes an explosion proof box, a linear CCD camera, two white LED light sources and a rotating motor. As the detonator is nearly cylindrical, a traditional CCD camera cannot acquire the entire surface image of the detonator at one time. A rotating motor drives a linear CCD camera and two LED light sources around the detonator. Then the detonator images are acquired. To avoid that detonator coded characters are not entirely captured, the rotating motor drives around the detonator in slightly more than one circle. By controlling the operating time of the motor, we can obtain the detonator images with fixed heights and widths. Thus, the sizes of the acquired detonator images are 320×640. Here, a portion of the surface of the detonator is repeatedly captured. It indicates that two identical detonator character strings may exist in one image. Therefore, a

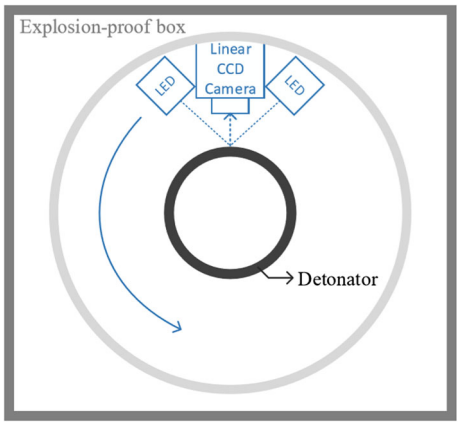

Fig. 1 Image acquisition system for detonators

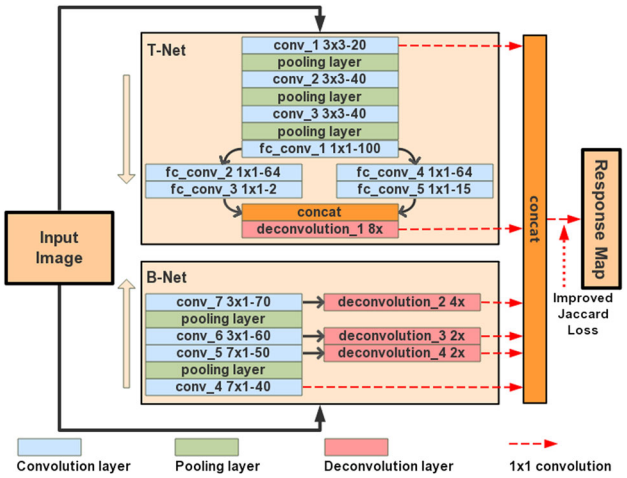

Fig. 2 Architecture of the detection network

post-processing scheme is employed to extract only one ROI of the detonator image, which will be discussed in Sect. 3.3.

3 Proposed method

3.1 The architecture of the proposed spotting system

The proposed spotting system includes a multi-scale detection network, a post-processing layer and a multi-label recognition network. An improved FCN including two subnetworks is designed as the detection network to predict the response map of the detonator image. The response map is processed by the post-processing layer to achieve the ROI involving the detonator coded characters. Finally, the detonator coded characters are recognized by the recognition network whose inputs are the ROIs of the detonators.

3.2 The detection network

The FCN can make pixel-level predictions and get the confidence indicating whether the pixels belong to the foreground. The response map involving the confidence can reflect the position of the text in the image [11]. Thus, inspired by the FCN, an improved FCN is designed as the detection network to generate an accurate response map of the detonator image in a coarse-to-fine mode (shown in Fig. 2). Different from the existing FCNs, the proposed FCN has two subnetworks called T-Net and B-Net to attempt to deal with the detonator images with low contrast. The B-Net is a CNN with some skip connections, which achieves a coarse response map. The T-Net is designed to distinguish the background similar to the text due to its character-level classification capabilities. By combining the T-Net and the B-Net with a concat layer and the IJLF, a final fine-tuned response map can be achieved to characterize the accurate position of the detonator coded characters. The input and the output of the detection network are an entire image and the corresponding response map, respectively. Some parameters of the layers are labeled in the layers. For each convolution layer, its parameters include name, kernel size and the total number of output feature maps. For example, the layer conv_1 uses a 3×3 convolution kernel with 20 output feature maps. For each deconvolution layer, its parameters include name and magnification.

(1) B-Net

The B-Net in the detection network has four convolutional layers and two pooling layers. In CNNs, it is necessary to integrate all the convolutional layers together to get a meticulous response map since local detailed information and global location information are extracted via the shallow layers and the deep layers, respectively [11]. Therefore, the four convolutional layers in the B-Net are concatenated together after deconvolutions and 1×1 convolutions. To keep the sizes of the feature maps equal to that of the input image, an upsampling operation may be implemented by each deconvolution layer. Among the four convolutional layers, the first convolutional layer does not need an upsampling operation, because the size of the feature map achieved by the layer is already equal to that of the input image. Other convolutional layers perform the $2\times$, $2\times$ and $4\times$ upsampling operations according to the sizes of the feature maps. Similar to [9], the convolutional kernels with large aspect ratios such as 7×1 and 3×1 are employed.

(2) T-Net

The T-Net, which uses the softmax loss functions for pre-training, performs a multi-label classification task to fine-tune the coarse response map achieved by the B-Net.

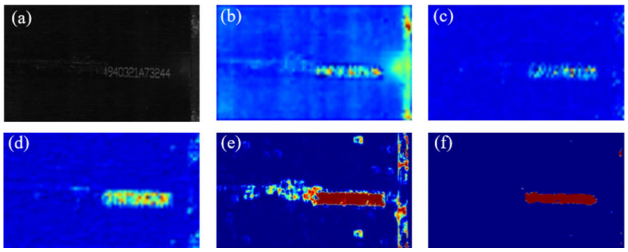

Fig. 3 Response maps obtained by different networks. **a** Input detonator image. The response maps obtained by **b** the 2-class branch in the T-Net; **c** the 15-class branch in the T-Net; **d** the T-Net; **e** the B-Net; **f** the detection network integrating the T-Net and the B-Net

It is implemented by a multitask learning scheme with two small branches. One performs a 2-class classification (text/non-text). The other performs a 15-class classification (0–9/A/H/X/S/non-text). Each training sample has two labels at the same time, for example, {text, 8} and {non-text, non-text}. The first and second labels indicate the 2-class and 15-class labels, respectively. During training, the first and second labels supervise the optimization process of the T-Net. As indicated in Fig. 3b–d, this helps the network to exclude some non-text noise similar to the text.

Due to the strategy based on these two labels, a challenge occurs during training, which is an imbalance between the total numbers of the text/non-text samples. For example, if the ratio of the total number of text samples to that of non-text samples is 1:1, the total number of non-text samples will be about 14 times more than that of the text samples for each character. This imbalance results in the final prediction with a large bias toward the same category. To solve this problem, a weighted softmax loss function is used to train the 15-class branch, which is defined as

$$L_s = \begin{cases} -\frac{\omega}{N_n}\sum_i \log\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}, & y_i \text{ is non - text} \\ -\frac{1}{N-N_n}\sum_i \log\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}, & \text{others} \end{cases} \tag{1}$$

where N is the total number of training samples, N_n is the number of non-text training samples, y_i is the second label of the ith sample, f_j is the jth element in the vector of the class scores and ω is a weight for performing data balance. Here, ω is set to 1/14 as discussed above. All the fully connected layers in the T-Net are converted to the convolutional layers. Just like those in the B-Net, the feature maps are upsampled 8× after concatenating two small branches.

(3) Integration of the T-Net and the B-Net

The outputs of different networks are separately shown in Fig. 3b–f. It can be seen that the response maps generated by the networks with the 2-class branch and the 15-class branch are not so accurate as the entire T-Net. Also, the T-Net achieves a weaker response than the B-Net although it can approximately locate the text region. This weak response will result in the failure of determining the accurate text region. Conversely, although the B-Net has a strong response to the text region, it is extremely sensitive to the noise in the background, which influences the precise location of the text region. Compared with the individual T-Net and the individual B-Net, the entire detection network can obtain an excellent response to the text region with extremely less noise (shown in Fig. 3f). The integration is simply illustrated in Fig. 2. Here, a concat layer is employed to concatenate six feature maps. Next, the six feature maps are integrated into a response map via a 1 ×1 convolutional operation and a sigmoid activation function. Thus, the values of the elements of the response map are normalized to [0,1]. At the training stage, the normalized response map and the ground truth are employed to calculate the loss via an IJLF. At the testing stage, the normalized response map of a detonator image is achieved by the trained detection network, which is subsequently transferred to the post-processing layer.

(4) Improved Jaccard loss function

The Jaccard loss function (JLF) optimizes the Jaccard index directly, which empirically works better than the weighted sigmoid cross-entropy. Suppose that 1 and 0 represent the ground-truth foreground and background, respectively; then, JLF is defined as [13]

$$L = 1 - \frac{card\left(Y_t \cap \hat{Y}_t\right)}{card\left(Y_t \cup \hat{Y}_t\right)} = 1 - \frac{\sum_{t\in Y_t}\left(1 \wedge \hat{y}_t\right)}{card(Y_t) + \sum_{n\in Y_n}(0 \vee \hat{y}_n)} \tag{2}$$

where Y_t and Y_n are the ground-truth foreground and background pixels, respectively. $\hat{Y}_t$ and $\hat{Y}_n$ represent the predicted foreground and background pixels, respectively. $card(.)$ represents the cardinality of a set. $\wedge$ and $\vee$ represent the logical AND/OR operations, respectively. Thus, $\hat{y}_t \in \hat{Y}_t$ and $\hat{y}_n \in \hat{Y}_n$. Since the elements of $\hat{Y}_t$ and $\hat{Y}_n$ are the probabilities whose values vary from 0 to 1, the JLF can be approximated by

$$\tilde{L} = 1 - \frac{\sum_{t\in Y_t}\hat{y}_t}{card(Y_t) + \sum_{n\in Y_n}\hat{y}_n} \tag{3}$$

It can be updated by

$$\frac{\partial \tilde{L}}{\partial \hat{y}_j} = \begin{cases} -\frac{1}{card(Y_t)+\sum_{n\in Y_n}\hat{y}_n}, & j \in Y_t \\ \frac{\sum_{t\in Y_t}\hat{y}_t}{(card(Y_t)+\sum_{n\in Y_n}\hat{y}_n)^2}, & j \in Y_n \end{cases} \tag{4}$$

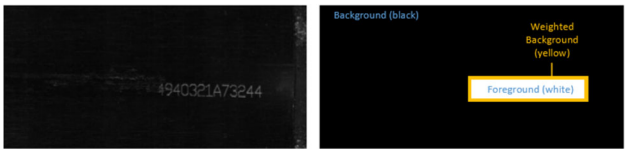

Fig. 4 An original detonator image (left) and its ground truth for performing text detection (right)

where j denotes the jth pixel in the input image and $\hat{y}_j$ represents the corresponding predicted value. To suppress the responses to the background around the text ROI, an IJLF is proposed by adding a regularization item into the JLF, which is defined as

$$\tilde{L}_{\text{imp}} = 1 - \frac{\sum_{f\in Y_t} \hat{y}_f}{card(Y_t) + \sum_{n\in Y_n} \hat{y}_n + \lambda \sum_{w\in Y_w} \hat{y}_w} \tag{5}$$

$$\frac{\partial \tilde{L}_{\text{imp}}}{\partial \hat{y}_j} = \begin{cases} -\frac{1}{card(Y_t)+\sum_{n\in Y_n} \hat{y}_n+\lambda \sum_{w\in Y_w} \hat{y}_w}, & j \in Y_t \\ \frac{\sum_{t\in Y_t} \hat{y}_t}{(card(Y_t)+\sum_{n\in Y_n} \hat{y}_n+\lambda \sum_{w\in Y_w} \hat{y}_w)^2}, & j \in Y_n \\ \frac{\lambda \sum_{t\in Y_t} \hat{y}_t}{(card(Y_t)+\sum_{n\in Y_n} \hat{y}_n+\lambda \sum_{w\in Y_w} \hat{y}_w)^2}, & j \in Y_w \end{cases} \tag{6}$$

where $\lambda \sum_{w\in Y_w} \hat{y}_w$ is a regularization term to regularize the influence of undesired responses. Y_w is the set of weighted ground-truth background pixels, which is a subset of Y_n. $\hat{y}_w$ is the predicted value of the element of Y_w. λ is a weight adjusting the penalties for the undesired responses around the ground-truth text. Here, $\lambda \geq 1$. The IJLF in (5) is the original JLF if $\lambda = 1$. The larger value of λ will result in the few occurrences of undesired responses. However, too large value of λ will erode the desired responses. Here, λ is set to 3 empirically.

An example is given to illustrate the relationship between the ground truth of a detonator image and its weighted background obtained by $\lambda \sum_{w\in Y_w} \hat{y}_w$ (shown in Fig. 4). The ground truth includes the text foreground shown by the white region, the non-text background shown by the black region and the weighted non-text background shown by the yellow region, which correspond to Y_t, Y_n and Y_w in (6), respectively. Here, the yellow region has a greater weight compared with the black region. Note that different colors are used only for the visualization purpose. In fact, the values of the pixels in the text/non-text regions are 1 and 0, respectively.

3.3 The ROI obtained by the post-processing layer

The response map achieved above is post-processed to obtain the ROI of the detonator image via the post-processing layer followed by the detection network. The input and the output of the layer are the response map and the desired ROI, respectively. To adaptively extract the ROI, a simple post-processing scheme is proposed as follows.

1. Binarization. As shown in Fig. 3, the differences between the foreground (text) and the background of the response map are quite significant. Thus, the OTSU method [14] can be employed to binarize the response map.
2. Connected component analysis (CCA). The CCA is employed to find all the connected components from the binarized response map.
3. Filtering strategy. A detonator has only one unique code. However, more than one connected component may be extracted from the binarized response map due to the mechanism of the image acquisition system and the existence of the background noise. Therefore, a filtering strategy is employed to determine the desired ROI. In practice, all the coded character strings in the detonators are of the same size. There are 13 characters in total and the size of each character is predefined. Assume that there are K connected components in the binarized response map; then, a confidence is assigned to each connected component, which is defined as

$$s_k = e^{-\frac{1}{2m_k}\left(\frac{|w_k-\bar{w}_g|}{w_k}+\frac{|h_k-\bar{h}_g|}{h_k}\right)}, \quad k = 1, 2, \ldots, K \tag{7}$$

where w_k and h_k are the width and height of the kth connected component in the binarized response map, respectively. $\bar{w}_g$ and $\bar{h}_g$ are the average width and height of the ground-truth ROIs for all the training detonator images, respectively. Here, $\bar{w}_g$ and $\bar{h}_g$ are set to 258 and 35, respectively. $m_k = \frac{1}{N_k}\sum_{l=1}^{N_k} \hat{y}_l$ is the average response intensity of the kth connected component, where N_k is the total number of pixels in the kth connected component. $\hat{y}_l$ represents the predicted probability of the lth pixel, whose value varies from 0 to 1. Therefore, $s_k \in (0, 1]$. The higher the value of s_k is, the more probably the corresponding connected component is found in the desired ROI. Thus, the connected component with the highest confidence is selected as the desired ROI of the detonator image.

3.4 The recognition network

A multi-label network is designed to recognize the detonator coded characters in the desired ROI, which is based on the network in [15]. The architecture of the recognition network is described as conv(5 ×5, 32)-pooling-conv(5 × 5, 64)-pooling-conv(3 ×3, 128)-pooling-conv(3 ×3, 256)-pooling-conv(3 ×3, 512)-fc(512)-13 ×fc(14), where conv, pooling and fc denote convolution layers, pooling layers and fully connected layers, correspondingly. The numbers in the parentheses indicate the attributes of corresponding layers. The input of the network is the desired ROI scaled to 35 ×258. Since the detonator coded character string has

a fixed size with 13 characters, the recognition network uses a single CNN structure followed by 13 independent softmax classifiers. The recognition network is trained with a general softmax loss function. Each classifier only predicts the character in the corresponding position of the coded string in order. The output of each classifier is a probability distribution of 14 classes (0–9/A/H/S/X). The class with the highest probability is considered as the recognition result of the corresponding classifier. This classification scheme has no annotations of locations and segmentation information.

4 Experimental results and discussions

In total, 33,647 detonator images acquired by the image acquisition system are involved in this section, which are randomly divided into two datasets. One is the training dataset with 30,000 samples. Another is the test dataset with 3647 samples. The source code of this paper is available at: https://github.com/CenGuandong/DCC.

4.1 The training strategy

The proposed spotting system is implemented by using Caffe [16]. Stochastic gradient descent (SGD) and the backpropagation (BP) algorithm are used to update the weights of each layer via a linear combination of the negative gradient and the previous updated weights. Followed by each convolutional layer, a rectified linear function ReLU is used as the activation function. All the weights are initialized with the Gaussian distribution whose mean value and variance are 0 and 0.01, respectively. The training strategy of the spotting system is described as follows.

Step 1: Pre-training the T-Net in the detection network. To train the T-Net, we randomly cropped 15,000 positive (text) samples and 15,000 negative (background) samples from the detonator images in the training dataset. Then, we annotated them with the labels (0–9/A/H/X/S/non-text). The size of each cropped sample is 35×18, which is equal to that of one-character region. When the pre-training is completed, all the fully connected layers are converted to 1×1 convolutional layers (fc_conv_x layers in Fig. 2, $x = 1, 2, 3, 4, 5$).

Step 2: Training the detection network integrating the T-Net and the B-Net. The pre-trained T-Net is used to initialize the upper part of the detection network. Then, the response map output by the entire detection network with the IJLF is approximated to the ground truth. Please refer to Sect. 3.2 for the training strategy of the detection network in detail.

Step 3: Training the recognition network. All annotated ROIs are cropped from the detonator images to train the recognition network. All the cropped ROIs are scaled to 35×258 and annotated with 13 character-level labels. Then, the recognition network is trained by the training strategy discussed in Sect. 3.4.

Step 4: Fine-tuning the entire spotting system. The parameters of the system are initialized by the independently trained detection and recognition networks. The input and the output of the system are the original detonator images in the training dataset and the corresponding recognized detonator coded character strings, respectively. The recognition network can adapt to the possible position deviations from the detection network via the post-processing layer.

For each network, the weight decay is set to 0.005. The learning rate is set to 10^{-4} and multiplied by 0.1 after each epoch. (One epoch involves 30,000 iterations.) The momentum is set to 0.9. For Steps 1–4, the batch sizes are 64, 1, 64 and 1, and there are 2, 4, 4 and 1 epochs, respectively. All the training processes are conducted in a computer with a NVIDIA Quadro K4200 4 GB GPU, an Intel Xeon E5-2630V3 2.4 GHz CPU and a 64 GB memory. However, it is worth noting that the test results are conducted by this computer without using the GPU.

4.2 Influence of the loss function on the performance of the detection network

In this section, the influence of the loss function on the performance of the detection network is discussed. Here, a metric named mean intersection over union (mIoU) [17] is used to evaluate the performance of the detection network. The larger the mIoU value is, the higher the quality of the generated response map is and the more excellent the detection performance is.

An experiment was conducted to investigate the influence of λ in (5) on the performance of the detection network. As shown in Fig. 5, the values of mIoU increase with the increase in the values of λ until $\lambda = 3$. However, the values of mIoU decrease with the increase in the values of λ when $\lambda > 3$. Therefore, λ is set to 3 empirically in this paper.

Another experiment was conducted to evaluate the influences of different loss functions, i.e., IJLF ($\lambda = 3$), original JLF [13] and weighted sigmoid cross-entropy loss function (WSCELF) [11, 18]. As illustrated in Fig. 6, the detection networks with three loss functions can output the response maps involving the approximate text regions. The sizes of the ROIs of the response maps are considered in the Jaccard-based

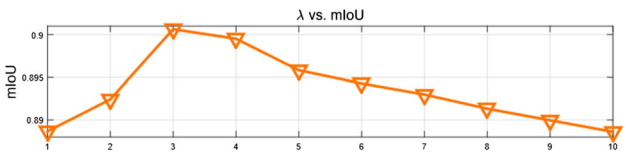

Fig. 5 Values of mIoU obtained by the detection network with different λ

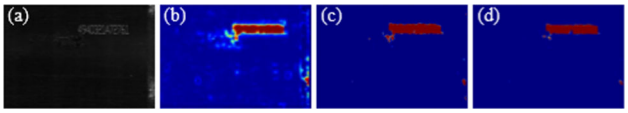

Fig. 6 Response maps obtained by different loss functions. **a** Original detonator image. **b** WSCELF. **c** Original JLF. **d** IJLF

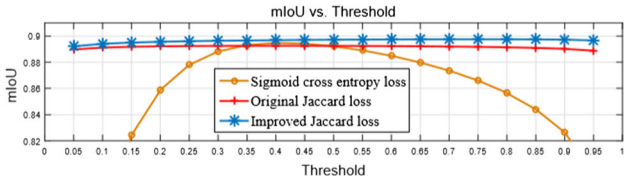

Fig. 7 Values of mIoU obtained by different loss functions with different binarization thresholds

Table 1 Values of mIoU obtained by different loss functions via the OTSU method

Loss functions	mIoU
WSCELF	0.894
JLF	0.889
IJLF	0.901

Table 2 Values of mIoU obtained by different training sets

Sets for training	Sets for verification	mIoU
1, 2, 3, 4	5	0.899
1, 2, 3, 5	4	0.900
1, 2, 4, 5	3	0.898
1, 3, 4, 5	2	0.901
2, 3, 4, 5	1	0.897

loss functions since a term $card(Y_t)$ is incorporated into the formulae. As indicated in (3) and (5), the gradients will be large if the ROIs are small. The detonator coded character region is a small region in the whole detonator image. Thus, the detection networks with the Jaccard-based loss functions can accurately approximate the desired response maps compared with the network with the WSCELF. Some undesired responses in the background around the ground-truth text may be obtained by the detection network with the original JLF (shown in Fig. 6c). The IJLF incorporates a regularization term to take into account the influence of the possible undesired responses. Therefore, the detection network with the IJLF can obtain the most excellent response map.

The threshold can influence the performance of the binarization method in the post-processing scheme. Subsequently, the extraction of text ROI may be influenced. We conducted an experiment to demonstrate that the detection network with the IJLF is not sensitive to the threshold of the binarization. As indicated in Fig. 7, the response map obtained by the detection network with the WSCELF is very sensitive to the threshold. This is because the log function $log(.)$ is used in the WSCELF. The values of $log(.)$ vary much faster when the values of function inputs are in the range from 0 to 0.5 compared with those in the range from 0.5 to 1. Hence, some pixels whose values are not closed to the label values (0 and 1) may appear in the response map. This implies that these pixels are hard to be identified as the text or the non-text pixels if the threshold is not appropriately selected. However, the values of mIoU obtained by the detection network with the Jaccard-based loss functions remain almost stable, indicating that the detection networks with the Jaccard-based loss functions are not sensitive to the threshold compared with that with the WSCELF. Furthermore, the detection network with the IJLF obtains a better response map than that with the original JLF. It is worth noting that the detection networks with three loss functions can obtain the similarly excellent response maps if the thresholds are elaborately selected.

For fair comparisons, the OTSU method is employed to binarize the response maps obtained by the detection networks with three loss functions. As indicated in Table 1, the detection network with the IJLF obtains the most excellent response map. However, the detection network with the original JLF obtains the worse response map than that with the WSCEL. Furthermore, as indicated in Fig. 7, the detection network with the WSCEL can obtain a greater value of mIoU than that with the original JLF if an appropriate threshold is selected.

To verify the stability and reliability of the designed detection network, we conducted a fivefold cross-validation experiment. We randomly divided all the training samples into five equally sized parts, which were labeled 1, 2, 3, 4 and 5, correspondingly. Four of them were used to train the detection network and one for verification every time. As shown in Table 2, there is no distinct fluctuation in mIoU, which indicates the stable and reliable performance of the detection network.

4.3 The comparisons with the state-of-the-art methods

In this section, the proposed spotting system is compared with some state-of-the-art deep learning-based methods, which are SSD [12], Textboxes [9], CTPN [10] and TextFCN [11]. It is well known that these state-of-the-art methods are excellent in text detection. For fair comparisons, the same recognition network discussed in Sect. 3.4 is used to recognize the detonator coded character strings in the ROIs achieved by these state-of-the-art methods. As discussed in Sect. 2, two identical detonator coded character strings may exist in one image. In real industries, only one detonator coded character string is recognized and the other one can be discarded. Therefore, the common metrics such as recall

Table 3 Comparison results achieved by different methods

Methods	Precision (%)	N_E/N_A	Accuracy (%)	Speed (s/image)
SSD [12]	94.269	337/3647	90.760	2.80±0.068
Textboxes [9]	98.163	270/3647	92.597	2.60±0.063
CTPN [10]	99.177	126/3647	96.545	16.75±0.424
TextFCN [11]	99.753	71/3647	98.053	9.09±0.229
Tiny TextFCN	96.298	358/3647	90.184	0.53±0.010
Ours (WSCELF)	99.452	115/3647	96.847	0.67±0.014
Ours (JLF)	99.808	111/3647	96.956	0.67±0.014
Ours (IJLF)	99.835	72/3647	98.026	0.67±0.014

and F-measure are not suitable here. To objectively evaluate these methods, two common metrics such as accuracy and precision are employed.

1. Accuracy. In this paper, the recognition of the detonator image can be considered as correct only if all the thirteen detonator coded characters in the image are identified correctly. Thus, the recognition accuracy is calculated as Accuracy $= (N_A - N_E)/N_A$, where N_E is the total number of the detonator images identified incorrectly and N_A is the total number of all test samples.
2. Precision. The precision represents the proportion of the total number of the ROIs detected correctly to the total number of extracted ROIs. It is defined as Precision $=$ TP/(TP + FP), where TP and FP are the total number of the precisely detected ROIs (IoU > 0.7) and the total number of the imprecisely detected ROIs, respectively.

As indicated in Table 3, the detection and recognition results obtained by the SSD are not satisfactory compared with other methods although it has a low computational complexity. This is because the SSD is designed for general object detection rather than for text detection. On the other hand, the Textboxes is elaborately designed for text detection. It modifies some parameters of the SSD according to the characteristics of the texts. Hence, compared with the SSD, the Textboxes can detect the ROI of the detonator image more precisely and achieve a more accurate recognition of the detonator coded characters. Since the CTPN potentially incorporates the context information of each region in the LSTM layers, it can achieve a better performance than the original SSD and the Textboxes in terms of detection precision and recognition accuracy. However, it has a very heavy computational burden since the LSTM and the sliding window schemes are time-consuming. Furthermore, a bottom-up approach to composing a text line influences its operation speed. These facts indicate that the CTPN cannot be employed in real industries. Also, its performance is not so meticulous as the pixel-level classification methods such as the TextFCN.

The TextFCN integrates all the features potentially extracted by every layer to obtain the semantic features. Therefore, it can do an excellent job of detonator coded character spotting at the recognition accuracy of 98.053%. However, it also achieves a poor performance in terms of computational burden when it is implemented in CPU. This is because there are too many layers and parameters in each layer of the network and the feature maps achieved by the layers are integrated together to obtain the semantic features. To fasten the TextFCN, we reduced the total number of layers and the parameters of each layer and designed a Tiny TextFCN. Although the Tiny TextFCN satisfies the efficiency requirement of real industries such as the processing speed up to 0.53 s/image, its performance of detonator coded character recognition remarkably falls. It is even worse than that of the original SSD. Compared with the TextFCN, the detection network of the proposed spotting system has fewer layers and fewer parameters in each layer. It employs a fine-tuning scheme via the T-Net to obtain the response map. Thus, the proposed spotting systems with different loss functions obtain similarly excellent detection and recognition results compared with the TextFCN. However, they require few computational burdens in CPU than the TextFCN. They can recognize a detonator image in only 0.67 s via CPU. This indicates that the proposed spotting system is suitable for real industries. On the other hand, the proposed spotting system with the IJLF achieves the best performance in terms of detection and recognition among all the methods. This fact also verifies the discussions on the influences of loss functions in Sect. 4.2.

5 Conclusions

This paper designs an image acquisition system to safely achieve the detonator images and proposes a text spotting system for detecting and recognizing detonator coded characters. The spotting system includes a multi-scale detection network, a multi-label recognition network and a post-processing layer. In the detection network, a well-designed

FCN is proposed to obtain a response map of the text region. The proposed FCN contains two subnetworks called B-Net and T-Net. The B-Net fuses the information of multiple layers to achieve a coarse response map. The T-Net performs a multi-label classification task to fine-tune the coarse response map. Two branches in the T-Net perform 2-class classification (text/non-text) and 15-class classification (0–9/A/H/X/S/non-text), respectively. An IJLF is defined to train the entire detection network, which adds a regularization term into the original JLF. A post-processing layer is used to post-process the response map obtained by the detection network so that the ROI of the detonator image is obtained. Finally, the detonator coded characters in the image are recognized when the ROI is fed into the recognition network. The recognition network is a CNN with the parallel classifiers so that it can classify every character in the ROI without supervising the location and the segmentation information. The experimental results indicate that the proposed spotting system achieves a better performance than the state-of-the-art methods for detecting and recognizing detonator coded characters. On the other hand, our proposed spotting system achieves an excellent performance in terms of the required computational burden implemented in CPU. This fact indicates that the proposed system is suitable for performing detonator coded character spotting in real industries.

As discussed in Sect. 4.2, the value of λ is empirically determined for detonator coded character detection. Although this empirical value of λ can be also directly applied to other similar tasks, the text regions may be not precisely detected. That is to say, its value had better be adjusted for other tasks. In the future work, we will propose an adaptive scheme to determine the value of λ so that our proposed method can be directly applied for similar tasks with no parameter justifications.

Acknowledgements This work was in part supported by the National Natural Science Foundation of China (Grant No. 61001179) and the Key Project of Industry-University-Research Collaborative Innovation in Guangzhou, China (No. 201802020010).

References

1. Sun, M., Zhang, L., Gao, H.: The design of recognition system for numbers of detonators based on support vector machine. J. Wuhan Univ. **59**(3), 245–248 (2013)
2. Zhu, A., Wang, G., Dong, Y.: Robust text segmentation in low quality images via adaptive stroke width estimation and stroke based superpixel grouping. In: 12th Asian Conference on Computer Vision, pp. 119–133. Springer (2014)
3. Gopalan, C., Manjula, D.: Statistical modeling for the detection, localization and extraction of text from heterogeneous textual images using combined feature scheme. SIViP **5**(2), 165–183 (2011)
4. Nirmala, S., Nagabhushan, P.: Foreground text segmentation in complex color document images using Gabor filters. SIViP **6**(4), 669–678 (2012)
5. Hadjadj, Z., Cheriet, M., Meziane, A., Cherfa, Y.: A new efficient binarization method: application to degraded historical document images. SIViP **11**(6), 1155–1162 (2017)
6. Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: 21st International Conference on Pattern Recognition (ICPR2012), pp. 3304–3308. IEEE (2012)
7. Jaderberg, M., Vedaldi, A., Zisserman, A.: Deep features for text spotting. In: 13th European Conference on Computer Vision, pp. 512–528. Springer (2014)
8. He, T., Huang, W., Qiao, Y., Yao, J.: Text-attentional convolutional neural network for scene text detection. IEEE Trans. Image Process. **25**(6), 2529–2541 (2016)
9. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: A fast text detector with a single deep neural network. In: 31st AAAI Conference on Artificial Intelligence, pp. 4161–4167. AAAI (2017)
10. Tian, Z., Huang, W., He, T., He P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: 15th European Conference on Computer Vision, pp. 56–72. Springer (2016)
11. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4159–4167. IEEE (2016)
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: 15th European Conference on Computer Vision, pp. 21–37. Springer (2016)
13. Cai, J., Lu, L., Xie, Y., Xing, F., Yang, L.: Improving deep pancreas segmentation in CT and MRI images via recurrent neural contextual learning and direct loss function. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 674–682. Springer (2017)
14. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (2007)
15. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv: 1406.2227 (2014)
16. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2014)
18. Xie, S., Tu, Z.: Holistically-nested edge detection. In: IEEE International Conference on Computer Vision, pp. 1395–1403. IEEE (2015)