Electronic Engineering and Informatics
G. Izat Rashed (Ed.)

doi:10.3233/ATDE240101

Character Recognition of Low Illumination Product Marking Based on Vision

Zelin SHEN [a], Boyao CHEN [b], Yiming FAN [b], Xiaofeng LU [b1]
[a] *Montverde Academy Shanghai, No.91 Jianhao Road, Shanghai, China*
[b] *School of Communication & Information Engineering, Shanghai University, No.99 Shangda Road, Shanghai, China*

Abstract. This paper is mainly aimed at recognition of the horizontal indeterminate-length marking characters of products under low illumination conditions. We first obtain images to be processed from local galleries or the camera. Then, we apply multi-scale Retinex algorithm in HSV space to improve the brightness of images and denoising technology to improve the quality of images. Finally, we realize text detection through CTPN network model and character recognition through CRNN network model. End-to-end character recognition is achieved. The experimental results show that the accuracy of the CTPN network model is 80.91%, and the accuracy of the CRNN network model is 61.79% respectively.

Keywords. Character recognition, CTPN, CRNN

1. Introduction

Nowadays digital images have become a widely used media form. In order to improve the quality and visual effects of images, image enhancement is required to convert digital images into a form that is more suitable for processing and analysis by machines or humans. The traditional image enhancement technology mainly includes two kinds of image enhancement algorithms, spatial domain and frequency domain, which mainly analyze and process image grayscale images [1]. With the continuous development of image processing technology, scholars put forward new requirements for image processing methods, which require image enhancement and color restoration for color images. However, in many real life and natural scenes, due to the influence of factors such as shooting angle, shooting environment and equipment, a large number of images often have poor visual perception due to problems such as lighting and exposure, which reduces the image quality and its application value [2].

In order to improve the image quality of low-light images, scholars at home and abroad have proposed a variety of low-light image enhancement methods in recent years. Jobson et al. first proposed the Retinex algorithm, which conforms to the color constancy theory of the human eye and mainly enhances grayscale images [3]. In recent years, the Retinex algorithm has developed from single-scale to multi-scale, and then to a multi-scale algorithm that supports color restoration [4]. However, the traditional Retinex algorithm is prone to lose edge detail information and produce false halos, so scholars

[1] Corresponding author: Xiaofeng LU, School of Communication & Information Engineering, Shanghai University, e-mail: luxiaofeng@shu.edu.cn

have proposed a variety of methods to effectively improve the traditional Retinex algorithm. Liu et al. proposed a fast algorithm based on Retinex, which can enhance low-illuminance images and restore information lost by low-illumination [5]; Wang et al. proposed a low-illuminance color image enhancement algorithm based on Gabor filter and Retinex theory, the restored color information in the processed image is closer to the original image [6].

In recent years, character recognition algorithms based on deep learning have mainly focused on the research of scene text detection algorithms. The mainstream text detection algorithms at this stage can be divided into the following three categories: the first category is text detection algorithms based on text box regression, which can be subdivided into "top-down" detection algorithms such as Faster-RCNN, YOLOv3 [7] etc. with “bottom-up” detection algorithms such as Connected Text Proposal Network (CTPN) [8] etc. The second category is segmentation-based text detection algorithms. The advantage of this algorithm is that it does not need to consider the characteristics of long text and deformed text, but its disadvantage is that the initial result of segmentation is likely to cause adjacent text lines to stick together [9]. The third category is the regression-segmentation hybrid detection method. Segmentation-based text detection methods have outstanding performance in this regard. At present, most methods use detection first and then segmentation as the main process.

Traditional text recognition algorithms mostly use template matching as the core idea. Deep learning algorithms can transform text recognition into sequence learning problems, thus giving birth to end-to-end OCR technology based on deep learning. In recent years, the mainstream end-to-end text recognition technology is mainly based on CRNN model, RARE model, etc. The recognition algorithm based on the CRNN model can be subdivided into the CRNN+CTC [10] framework and the CNN+Seq2Seq+attention framework. The RARE model is mainly used for the recognition of curved and perspective text. In this model, the input image is first processed in the spatial transformation network (STN), and then the corrected image is sent to the sequence recognition network (SRN) to obtain the text prediction result.

Based on the above analysis, in order to realize the recognition of the horizontal indeterminate-length marking characters of products, the original image is obtained by collecting images in real time through the camera or calling the local library; in the HSV space, the multi-scale Retinex algorithm is used to improve the image brightness, and the denoising technology is used to improve the image quality; The CTPN network model performs text detection, the CRNN network model performs text recognition, and finally realizes the recognition of the horizontal indeterminate-length marking characters of products.

2. Image Preprocessing

In order for the character recognition model to accurately detect and recognize image text information, the input image must reach a certain degree of clarity, such as the brightness difference between the text and the background, the signal-to-noise ratio of the image, the matching of the text style with the detection model, and the incomplete text information. Therefore, before character recognition, images need to be preprocessed to attenuate interference noise, enhance useful information, and improve text detectability.

2.1 Realization and Analysis of Color Enhancement Based on HSV Space

In this paper, based on the HSV space, the adaptive multi-scale Retinex algorithm is used to enhance the color of the image. First, the multi-scale Retinex algorithm is used to process the V component of the image, and its calculation formulas are shown in Eq. (1) to Eq. (3). Among them, $S(x,y)$ is the input image, $F(x,y)$ is the center surround function, $R_{MSR}(x,y)$ is the output image, λ is a coefficient, k is the number of Gaussian center surround functions, and c is the Gaussian surround scale. In order to ensure that the output has the advantages of the high, medium and low scales of the single-scale Retinex algorithm, this experiment takes $K = 3$, $w_1 = w_2 = w_3 = 1/3$, and the corresponding Gaussian surround scale c takes 15, 80, and 200 respectively.

$$R_{MSR}(x,y) = \sum_k^K w_k\{logS(x,y) - log[F_k(x,y) * S(x,y)]\} \tag{1}$$

$$F(x,y) = \lambda e^{\frac{-(x^2+y^2)}{c^2}} \tag{2}$$

$$\iint F(x,y)dxdy = 1 \tag{3}$$

Then, adaptively adjust the results processed by the multi-scale Retinex algorithm, which implements references [11]. Calculate the mean(*mean*) and variance(*var*) of the V channel pixel, and obtain *Min* and *Max* according to the Eq. (4), where the parameter D is used to control the image to achieve dynamic adjustment without color shift; then $R_{MSR}(x,y)$ according to the Eq. (5) linear mapping, adding overflow judgment at the same time.

$$Min = Mean - D * Var \ , \ Max = Mean + D * Var \tag{4}$$

$$R_{autoMSR} = \frac{R_{MSR}(x,y) - Min}{Max - Min} * 255 \tag{5}$$

2.2 Realization of Image Denoising and Analysis of Results

Imaging in industrial environments tends to contain Gaussian noise, and the *GaussianBlur* function in the OpenCV library is used to remove Gaussian noise. The Gaussian filter is a linear smoothing filter mainly aimed at Gaussian noise. The definition of the Gaussian filter adopts a two-dimensional Gaussian distribution. The mathematical definition of Gaussian filtering on a two-dimensional plane is shown in Eq. (6) to Eq. (7).

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(\frac{x^2+y^2}{2\sigma^2})} \tag{6}$$

$$f_{output}(x,y) = G(x,y) * f_{input}(x,y) \tag{7}$$

Among them, (x,y) is the pixel coordinates; σ is the standard deviation of the Gaussian function ; $f_{input}(x,y)$ is the output image; * means convolution operation. Since Ksize, the size of the Gaussian kernel, directly affects the clarity of the final image, a Gaussian kernel with a size of (5,5) is selected after multiple test analysis.

In addition, for salt and pepper noise in imaging, an adaptive median filter is used to process such noisy images, and its mathematical expression is shown in Eq. (8).

Among them, $f'(x,y)$ represents the calculated pixel value ; S_{xy} indicates the area where the window is located.

$$f'(x,y) = \text{median(s,t)} \in S_{xy}\{g(s,t)\} \tag{8}$$

3. Text Recognition

3.1 Text Detection and Its Realization Based on CTPN

This topic is mainly aimed at the detection and the recognition of the horizontal indeterminate-length marking characters of products. In order to achieve the purpose of end-to-end recognition, the Connection Text Proposal Network (CTPN) is adopted as the text detection algorithm. The algorithm uses the VGG16 network as the convolutional neural network to complete the feature extraction task; uses the long short-term memory network (LSTM) as the recurrent neural network to establish the spatial dependence of the model context.

This article refers to references [8] and implements the CTPN network model based on Pytorch. When determining the final coordinate information in the regression layer, in addition to determining the coordinate information of the anchors in the original image, it is also necessary to predict and adjust the coordinates of the boundary anchors according to Eq. (9) and Eq. (10). In addition, the model training loss function is set according to references [8], as shown in Eq. (11). Among them, $v = \{\nu_c, \nu_h\}$ is the predicted coordinate; $v^* = \{\nu_c^*, \nu_h^*\}$ is the actual coordinate; c_y^a and h^a are the center and height of the anchor's y-coordinate; c_y and h are the center and height of the predicted y-coordinate; c_y^* and h^* are the center and height of the actual y-coordinate. s_i is the predicted probability that anchor i is the actual text, $s_i^* = \{0,1\}$ is the actual value, v_j is the predicted y-coordinate associated with anchor j, and v_j^* is the actual y-coordinate.

$$\nu_c = (c_y - c_y^a)/h^a \ , \quad \nu_h = \log(h/h^a) \tag{9}$$

$$\nu_c^* = (c_y^* - c_y^a)/h^a \ , \quad \nu_h^* = \log(h^*/h^a) \tag{10}$$

$$L(s_i, v_j, o_k) = \frac{1}{N_s}\sum_i L_s^{cl}(s_i, s_i^*) + \frac{\lambda_1}{N_v}\sum_j L_v^{re}(v_j, v_j^*) \tag{11}$$

3.2 Realization of text recognition based on CRNN

In order to avoid the drawbacks of traditional character recognition algorithms that require text segmentation and realize end-to-end text recognition, a combination of Convolutional Recurrent Neural Network (CRNN) and Connectionist Temporal Classification (CTC) is used to realize text recognition. The network model consists of three parts: a CNN network as a convolutional layer, a bidirectional LSTM network as a recurrent layer, and a CTC algorithm as a transcription layer. This paper implements text recognition based on Pytorch, and the entire implementation flow chart is shown in Figure 1.

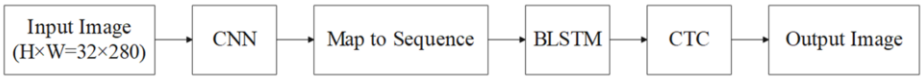

Figure 1. Flow chart of CRNN algorithm implementation

Among them, the Softmax function is used when the bidirectional LSTM outputs the predicted label distribution; when the CTC algorithm converts the predicted label distribution into the final label sequence, the "sequence merging mechanism" is used to avoid letter redundancy in the recognition result. Finally, the model training loss function is configured according to Eq. (12) and Eq. (13), x represents actual text, z represents the prediction result, S represents the set of all actual text characters, $B(\pi)$ represents the collection of all paths of the predicted results z of the actual text x.

$$L(S) = -\sum_{(x,z)\in S} \ln p(z|x) \tag{12}$$

$$p(z|x) = \sum_{\pi:B(\pi)=1} p(\pi|x) \tag{13}$$

4. Analysis of Results

4.1 Image preprocessing experiment results analysis

For the direction of experimental research, multiple pictures of marking characters of products were collected under backlight or poor lighting conditions. Perform color enhancement processing on it. The effect is shown in Figure 2, where the left side is the original image, and the right side is the color-enhanced picture. It can be seen that after processing with the algorithm of this paper, the image is not only much brighter and clearer from the subjective visual effect, but also has obvious changes from the computer vision aspect, and the subsequent text detection model can also correctly capture the text area in the image.

Figure 2. Original image and preprocessed image

4.2 CTPN model training and analysis of experimental results

The experimental environment of the CTPN model is NVIDIA GeForce GTX 1050 platform, using Torch1.8.1 as the backend, and Keras as the deep learning framework. There are 1434 training sets and 408 test sets, selected from ICDAR2013_Focused Scene Text, ICDAR2019_MLT, and ICDAR2015_Born-Digital Images. In this paper, the above training set is used to train the CTPN text detection model for 100 iterations, and

the training loss line chart shown in Figure 3 is obtained. Among them, the green curve is the final predicted anchor vertical coordinate offset regression loss, the blue curve is the text/non-text classification loss, and the red curve is the total loss.

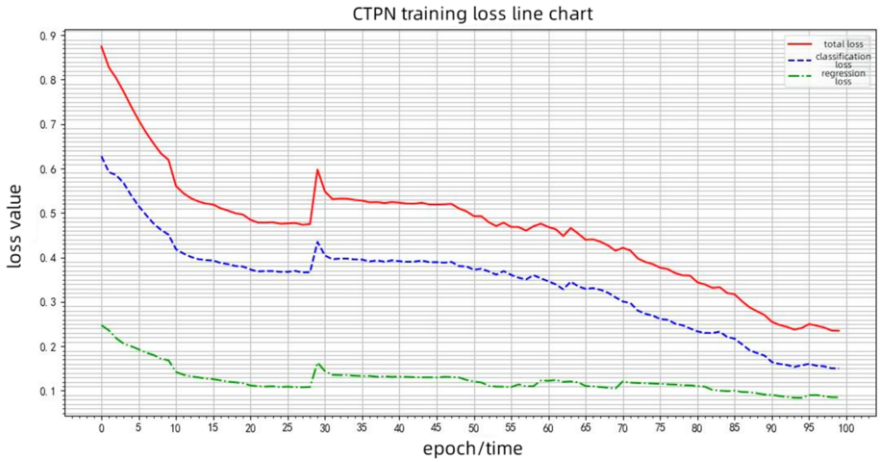

Figure 3. CTPN training loss line chart

Table 1. CTPN model test results

Method	IoU	Recall	H-mean	Precision
CTPN_ICDAR	0.5	55.34%	65.73	80.91%

We use the ICDAR2013_Incident Scene Text official end-to-end evaluation script to test the model with the test set. When the IoU threshold is 0.5, the P value of the model is 80.91%, the R value is 55.34%, and the F1 value is 65.73%, as shown in Table 1. Comparing the officially calibrated text area of the test set (left picture) and the predicted text area of the model (right picture), the model can correctly calibrate characters written horizontally and with normal text size, as shown in Figure 4.

Figure 4. CTPN model text area prediction results

4.3 CRNN model training and analysis of experimental results

The experimental environment of the CRNN model is the NVIDIA GeForce GTX 1050 platform, using Torch1.8.1 as the backend, and Keras as the deep learning framework. The data set used in this model is produced using the open source project "Paddle OCR_TextRender2.0". According to the needs of the project, the pictures imitate three situations: metal plate laser lettering, light background printing and dark background printing. In addition, the size of the data set is 32×280, training set: verification set: test set = 18468:5582:5261, about 6:2:2 distribution with no repetition. The sample images of the CRNN dataset are shown in Figure 5.

Figure 5. CRNN dataset sample picture

In this experiment, the model was trained for 84 iterations using the training set and the verification set. After adjusting the training parameters, the training loss and validation set accuracy line chart shown in Figure 6 was obtained.

Among them, the red curve is the training loss, and the blue dotted line is the validation set accuracy. The whole training process is divided into two parts. The learning rate is 0.0003 for the first 25 times, and then the learning rate is adjusted to 0.0001. After the training loss value drops smoothly to 0.13, it shows a convergence trend. Therefore, the 84th training result is taken as the final parameter, and the test set is used to test the model whose precision is 61.97%.

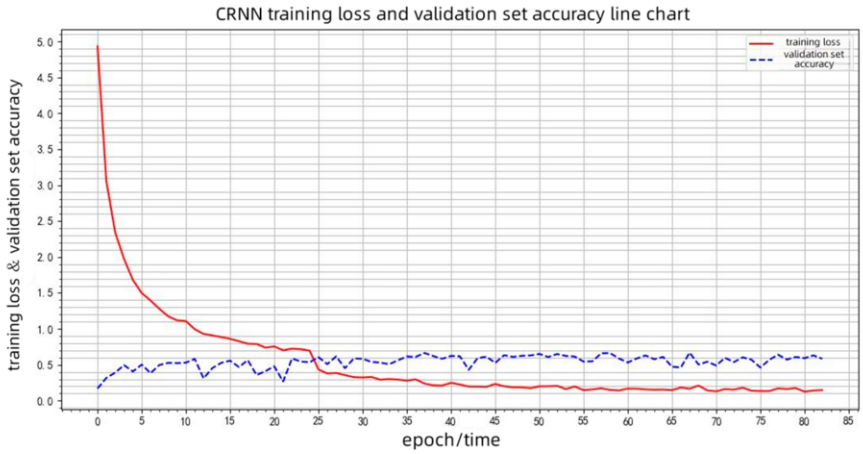

Figure 6. CRNN training loss and validation set accuracy line chart

5. Conclusion

In this experiment, a system that can perform image preprocessing, text detection and character recognition is designed. In the character recognition front-end, in order to make the image quality reach a clear and recognizable degree from the computer vision aspect, pre-processing operations such as multi-scale Retinex algorithm, adaptive median filtering and Gaussian filtering are executed. In addition, in order to get the text detection model and text recognition model matching the research direction of the subject, the corresponding data sets were selected and produced after experiments. After training and testing, the CTPN model with 80.91% accuracy in text detection and the CRNN model with 61.97% correctness in text recognition are obtained.

References

[1] Murthy K, Shearn M, Smiley B D, et al. (2014) SkySat-1: very high-resolution imagery from a small satellite. In: Sensors, Systems, and Next-Generation Satellites. Amsterdam. pp. 367-378. https://doi.org/10.1117/12.2074163.

[2] Cao N, Lyu S, Hou M, et al. (2021) Restoration method of sootiness mural images based on dark channel prior and Retinex by bilateral filter. Heritage Science, 9(1): 1-19. https://doi.org/10.1186/s40494-021-00504-5.
[3] Land E H, McCann J J. (1971) Lightness and retinex theory. pp. 61(1): 1-11. https://doi.org/10.1364/josa.61.000001.
[4] Rahman Z, Jobson D J, Woodell G A. (2004) Retinex processing for automatic image enhancement. Journal of Electronic imaging, 13(1): 100-110. https://doi.org/10.1117/1.1636183.
[5] Liu S, Long W, He L, et al. (2021) Retinex-based fast algorithm for low-light image enhancement. Entropy, 23(6): 746. https://doi.org/10.3390/e23060746.
[6] Wang P, Wang Z, Lv D, et al. (2021) Low illumination color image enhancement based on improved Retinex theory. In: 2020 4th International Conference on Electronic Information Technology and Computer Engineering. Shanghai. 334–339. https://doi.org/10.1145/3443467.3443777.
[7] Farhadi A, Redmon J. (2016) YOLO9000: better, faster, stronger. https://doi.org/10.48550/arXiv.1612.08242.
[8] Tian Z, Huang W, He T, et al. (2016) Detecting text in natural image with connectionist text proposal network. In: European Conference on Computer Vision 2016. Amsterdam. pp. 56-72. https://doi.org/10.1007/978-3-319-46484-8_4.
[9] LI Y, CHEN Y. (2021) Review on Deep Learning Based Scene Text Detection. Computer Engineering and Applications, 57(6):42-48. https://doi.org/10.3778/j.issn.1002-8331.2010-0468.
[10] Graves A, Santiago Fernández, Gomez F, et al. (2006) Connectionist temporal classification:labeling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. Pittsburgh. pp. 369-376. https://doi.org/10.1145/1143844.1143891.
[11] Jobson D J, Rahman Z, Woodell G A. (1997) A multiscale retinex for bridging the gap between color images and the human observation of scenes. IEEE Transaction on Image Processing, 6(7):965-976. https://doi.org/10.1109/83.597272.