Character Recognition for Automotive Parts Coding Based on Convolutional Neural Network

Daiyang Li\textsuperscript{1a}, Qian Tang\textsuperscript{1b,\textdagger}, Hao Zhou\textsuperscript{1c}, Yi Li\textsuperscript{1d}, Qiguang Su\textsuperscript{1e} and Zhaofu Lin\textsuperscript{2f}

\textsuperscript{1}State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, China; \textsuperscript{2}CHONGQING CHANGAN AUTOMOBILE Co., Ltd, Chongqing 400023, China

leedy1205@163.com, \textsuperscript{b,\textdagger}tqcqu@cqu.edu.cn, \textsuperscript{c}haozh@cqu.edu.cn, \textsuperscript{d}20142260@cqu.edu.cn, \textsuperscript{e}suqiguang666@gmail.com, \textsuperscript{f}linzf@changan.com.cn

Abstract. The recognition of coding characters is vital in automobile manufacturing and assembling, since it guarantees the circulation of essential information on production line. However, due to the fact in real world industrial applications, parts are involved in complex working situation such as reflection, smudge, abrasion and etc. Degradation of the performance of recognition is extremely serious. In this paper, a convolutional neural network (CNN) based method, including localization, segmentation and recognition is proposed to address the problem. Firstly, in order to meet the complex working situation as well as improving the stability and accuracy of character region localization, this method proposes an improved Maximally Stable Extremal Regions (MSER) algorithm introducing Gaussian distribution probability judgment and gray threshold judgment. Next, the combination of statistics and projection algorithm achieves single character segmentation which does not need any prior knowledge. Finally, a convolutional neural network is constructed to recognize character. The proposed method is evaluated through experiments on a platform. Experimental results and comprehensive comparison analysis with respect to traditional recognition method have demonstrated the superiority of the proposed method.

1. Introduction

The code on the parts contains a lot of information about the product, such as production date, production batch, and supply object. Companies choose to use product coding to ensure product tracking and traceability. In the process of coding, it is inevitable that there will be problems such as character painting errors and incomplete character painting. Therefore, the detection of the coding defects becomes a significant step in the manufacture of parts. Due to the shortcomings of low precision, high labor intensity and high cost of personnel, the coding character recognition technology based on computer vision has been becoming of great importance.

The key to coding character recognition is positioning, segmentation and recognition. A variety of image processing algorithms and intelligent recognition algorithms are applied in these three aspects. Based on the Maximally Stable Extremal Regions (MSER) \cite{1} algorithm, L Neumann \cite{2} proposes multi-stage complexity calculation to enhance the accuracy of character region recognition. T Wu \cite{3} and S Afroge \cite{4} use traditional methods for character segmentation and recognition, and verify the high recognition accuracy only for simple object. M Jaderberg \cite{5}, Z Tian \cite{6}, and B SHI \cite{7} have
achieved good results using deep learning methods, but rely on a great deal of training and high-intensity operations. For industrial production, high real-time performance is required. Slant characters, long text and point-like characters can affect the reliability and accuracy of these algorithms.

In order to address the problems above, this paper proposes a new character recognition method. In summary, the contributions of the paper are the following: 1) An improved Maximally Stable Extremal Regions algorithm with Gaussian distribution probability judgment and gray threshold judgment is proposed to deal with complex condition; 2) A character segmentation algorithm is proposed to achieve single character segmentation which does not need any prior knowledge; 3) A convolutional neural network is constructed to recognise characters. As shown in Figure 1.

The rest of the paper is organized as follows: The localization of the coding character region is introduced in Section 2. Section 3 presents the segmentation of the coding characters. And the recognition of the coding characters is provided in section 4. Some experiments are conducted to evaluate the proposed method against some other methods in Section 5. Section 6 concludes this paper and suggests topic for the future research.

2. Localization of the coding character region

2.1. Edge gradient enhancement
Gaussian filtering is able to blur the noise area and reduce interference to the region of interest (ROI). Meanwhile, the character region is inevitably affected by filtering. The accuracy of MSER [1] algorithm is quietly dependent upon the obviousness of both the character region gradient and the background surrounding. The Canny operator is used to detect the edge of the filtered image. By enhancing the gray value of the edge region of the character, the difference between the foreground and the background becomes significant (see Figure 2).

2.2. Localization of the character region
The MSER algorithm [1] performs multiple binarization processes on the entire image, and the threshold is taken from 0 to 255. In this process, as the binarization threshold rises, the change in the area of some connected areas is small. These areas are called the maximally stable extremal regions, i.e., MSER. For automotive parts coding images, the character area is a typical representation of
MSER.

The gray image $G$ is divided into $n$ connected regions in accordance with the gradation value of each pixel. Then the area of the connected region changes in the binarization threshold change $\nu(i)$ is given in (1):

$$\nu(i) = \frac{|S_i + \Delta - S_i - \Delta|}{|S_i|}$$

(1)

Where $S_i$ is the area of the $i$-th connected area, and $\Delta$ denotes the minor threshold change. When $\nu(i)$ is less than a given threshold, the area is considered to be the MSER of image $G$. Set $A$ be the set of MERS of image $G$, $A = \{A_1, A_2, A_3, ..., A_i\} (i = 1, 2, 3 ...)$. The connected domains of some non-character regions (such as stains, abrasions, reflections...) also belong to the MSER. These are seriously affecting the localization stability (see Figure 3). In this case, the elements in set $A$ need to be filtered.

![Figure 3](image)

**Figure 3.** Misrecognition of traditional MSER algorithm, (a) reflection and abrasion, (b) smudge, (c) reflection and smudge. Red circle indicates misrecognition area.

It is obvious that the character position is concentrated and the gray value of the character region is large. According to these characteristics, the probability judgment based on the Gaussian distribution and the gray threshold judgment are introduced into the MSER algorithm to finish the accurate localization of the character region.

Since the positions of the characters are relatively concentrated, the coordinates $(x_i, y_i)$ of the center point $C_i$ of the MSER set $A_i$ can be regarded as obeying the Gaussian distribution, $X \sim N(\mu_x, \sigma_x)$, $Y \sim N(\mu_y, \sigma_y)$. Where $\mu_x$ and $\mu_y$ is the mean of $x_i$ and $y_i$, $\sigma_x$ and $\sigma_y$ is the variance of $x_i$, $y_i$. Small probability events usually refer to events with a probability of less than 5%. For Gaussian distributions, $P[X - \mu < 2\sigma, \mu + 2\sigma] = 0.9544$. Thus, only the value in the range of $(\mu - 2\sigma, \mu + 2\sigma)$ is reserved while the area $A_i$ corresponding to the center point $C_i$ outside the range will be discarded. Then get the set $B$ that satisfy the condition, $B_i = \{A_i| x_i \in (\mu_x - 2\sigma_x, \mu_x + 2\sigma_x), y_i \in (\mu_y - 2\sigma_y, \mu_y + 2\sigma_y)\}$.

Let the set $P_i (i = 1, 2, 3 ...)$ be the set of points of the region $B_i \in B (i = 1, 2, 3 ...)$. For all points $p_k \in P_i (k = 1, 2, 3, ..., n)$ in each region $B_i$, the average gray value $\bar{f}_i$ as shown in (2)

$$\bar{f}_i = \frac{\sum_{k=1}^{n} f_{p_k}}{n}$$

(2)

In Eq (2), $f_{p_k}$ denotes the gray value of $p_k$. Given the threshold $t$, and eliminate the region $B_i$ in which $\bar{f}_i < t$. The final improved MSER algorithm result is shown in Figure 4.

![Figure 4](image)

**Figure 4.** The recognition result of improved MSER algorithm.
Locating every single character directly leads to a high error rate. Therefore, all the detected and retained MSERs are included in a rectangle to locate the entire character region. The localization effect is shown in Figure 5.

2.3. Character region correction
It is essential to perform the procedure of extracted coding area correction for the simple reason that keeping all codes horizontal is almost impossible in practice. Since the character is tilted, the coordinate values of the character region and the non-character MSER region become close, resulting in an increase in the probability estimation error based on the Gaussian distribution previously proposed. To address the problem, more stringent parameter settings (reduce $(\mu - 2\sigma, u + 2\sigma)$ to $(\mu - \sigma, u + \sigma)$) are adopted to locate a part of the characters. And then the affine transformation is performed by the positioned characters. After the correction is completed, most of the characters are located by normal parameters. The process is as shown in Figure 6.

3. Segmentation of the coding characters

3.1. Line segmentation
The code consists of two lines. The line segmentation is first performed to divide the entire coding area into two parts. A binarized image $I \in R^{\text{width} \times \text{height}}$ is obtained from the coding area. The number of white pixels in each row $\text{white}[i]$ as described in (3):

$$\text{white}[i] = \sum_{j=1}^{\text{width}} g(i, j)$$

Where $\text{width}$ is the total number of image pixels columns and $g(i, j)$ is the value of the binarized image at $(i, j)$.

In the coding area, the division line is located approximately in the middle of the image height as the heights of the two lines of code are almost the identical. Find the $\text{white}[i]_{\text{min}}$ in $\text{white}[0.4 \times \text{height}, 0.6 \times \text{height}]$, the corresponding $i$ is the split line. As shown in Figure 7.
3.2. Column segmentation

After line segmentation, column segmentation is used to get the individual characters. Vertical projection of a single-line character area results in Figure 8. Obviously, the coding characters are discontinuous due to the uneven surface. There are many columns inside the single character without white pixels, such as the portion circled in red. This has caused a serious interference with character segmentation. It is necessary to dilate the characters. The processed character image and projected image are as shown in Figure 9.

After dilating, the number of white pixels in each column \( \text{white}[j] \) is described in (4):

\[
\text{white}[j] = \sum_{i=1}^{\text{row}} g(i, j)
\]  

(4)

Where \( \text{row} \) is the total number of rows of image pixels. The result of conventional projection method is shown in Figure 10. A line of characters is divided into blocks of one or more characters. Set \( \text{block}[i] (i = 1, 2, 3...) \) be a set of the widths of all the character blocks divided by one line of characters.

As Figure 10 shows, it is clear that the traditional segmentation method is not effective as characters are adhered after dilating. In order to deal with this situation, a priori knowledge such as the number of characters and the actual width of characters is required. Y Nan [8] and M Pei [9] completed the division through these prior knowledges. When the number of characters or the actual width of the object changes, the parameters need to be set again. This type of segmentation method that relies on prior knowledge is highly reliable but less applicable.
It has been found through experiments that a part of a single character can be directly segmented after coarse segmentation. On this basis, this paper proposes to introduce statistical methods into the projection segmentation algorithm. The standard character width is determined based on the information provided by the image itself without prior knowledge.

For standard code characters, the width of a single character should be proportional to its height. The width is in the range of $[0.4 \times \text{row}, 0.8 \times \text{row}]$, and the row is known. So by counting the values in the range of $[0.4 \times \text{row}, 0.8 \times \text{row}]$ in $\text{block}[i]$, we get a set of single characters $\text{single}[i] = \{\text{block}[i] | \text{block}[i] \in [0.4 \times \text{row}, 0.8 \times \text{row}], i = 1, 2, 3 \ldots\}$. The standard width of a single character in this line is defined as (5):

$$\text{std} = \frac{\sum_{i=1}^{n} \text{single}[i]}{n}$$

(5)

If the coarse segmentation does not split a single character, the $0.6 \times \text{row}$ is used as the standard width value $\text{std}$. For the case where the character block width is greater than $1.5 \times \text{std}$, it is considered that at least two code characters have been conglutinated. Since the character widths are substantially the same, the split column of the concatenated characters is roughly in the area of $[0.8 \times \text{std}, 1.2 \times \text{std}]$ starting from the first column of the first character. According to the boundary feature of the adhesion region, the column with the smallest white pixel in the area can be used as the split column. Multi-character blocking can be split in the same way. In order to facilitate subsequent recognition, the segmented characters are normalized after the segmentation is completed. The size of the divided picture is adjusted to $28 \times 28$ pixels by linear interpolation. The segmentation result is shown in Figure 11.

![Figure 11. The result of segmentation.](image)

4. Recognition of the coding characters

The convolutional neural network is used to achieve the character recognition for the case that the code characters are discontinuous and the same character has a tilt or thickness change. This paper focuses on the part including 37 specific characters in which 10 digits from 0 to 9, English letters from A to Z, as well as "-". The convolutional neural network structure constructed in this paper has 6 layers, which are 2 convolutional layers, 2 pooling layers, 1 fully connected layer and output layer (see Figure 12).

![Figure 12. Neural network structure.](image)
4.1. Activation function
The convolutional layer, the sampling layer, and the fully connected layer can only perform linear operations. Neural networks cannot fully exploit their performance. An activation function that can perform nonlinear operations is needed to process the data. B Xu [10] proposed RReLu on the basis of ReLu [11], and the formula is (6):

\[ y_{ji} = \begin{cases} x_{ji} & x_{ji} \geq 0 \\ a_{ji}x_{ji} & x_{ji} < 0 \end{cases} \]

(6)

Where \( a_{ji} \sim U(l, u) \), \( l < u \), \( l, u \in [0,1) \).

Compared with other activation functions, RReLu has obvious advantages. The comparison of various activation functions is shown in Table 1.

| Function  | Vanishing gradient | Zero-centered | Power operation |
|-----------|--------------------|---------------|----------------|
| Sigmoid   | ✓                  | ✗             | ✓              |
| tanh      | ✓                  | ✓             | ✓              |
| Softplus  | ✓                  | ✗             | ✓              |
| ReLu      | ✓                  | ✗             | ✗              |
| RReLu     | ✗                  | ✓             | ✗              |

4.2. Optimization algorithm
The neural network can only exert its best effect when the weight parameters of each neuron are appropriate. These parameters need to be optimized through training. Therefore, the training speed and recognition accuracy is directly related to the optimization algorithm. D Kingma [12] proposed the ADAM algorithm based on Stochastic gradient descent (SGD) algorithm, and its expression is given in (7) (8) (9):

\[
\theta_{i+1} = \theta_i - \eta \frac{\hat{m}_i}{\sqrt{\hat{v}_i} + \epsilon} \\
\hat{m}_i = \frac{m_i}{1 - \beta_1^t} \\
\hat{v}_i = \frac{v_i}{1 - \beta_2^t} \\
m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \\
v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2
\]

(7) (8) (9)

Where \( \beta_1, \beta_2 \) is the attenuation coefficient, \( m_t, v_t \) respectively represent the mean of the gradient and the mean of the square of the gradient.

Using the datasets collected in this paper, several optimization algorithms [13] [14] were compared experimentally. The results are shown in Figure 13.
5. Experiments

5.1. Experimental Platform
The camera used in the experiment is: Jing Hang JHSM300f industrial camera, the sampling unit is 1/2 inch CMOS, and the image resolution is 800×600 pixels. PyCharm and opencv3 are used to carry out all the operations, running on an Ubuntu system with Intel I5-8300H (2.3GHz), and 8GB memory.

5.2. Results and analysis
Capture 100 photos by industrial cameras from different parts and different situations. In terms of character region localization and segmentation effects, the algorithm proposed in this paper can work on all 100 images collected. The average localization time of each image is 0.14s, while the CTPN [6] algorithm requires an average of 1.01s. All 2748 characters on the picture were successfully split into a single picture. The average processing time per character is approximately 0.02s.

There are 29 types of characters, including 10 types of numbers, 18 types of uppercase letters, and 1 type of special characters, since some letters are not used by the company. Various characters are divided into a training set and a test set according to the weight ratio. The training set has a total of 2198 characters, and the test set has a total of 550 characters.

The training set is put into the neural network for parameter training, and then the training effect is tested through the test set. The learning rate is set to 0.001, the batch size is set to 10, the epoch is set to 10. In order to compare the recognition effects of different classification methods, experiments were carried out in various classification methods using the same training set and test set. The experimental results are shown in Table 2.

Table 2. Comparison of different classification methods.

| Method               | Training cost(s) | Test cost(s) | Accuracy   |
|----------------------|------------------|--------------|------------|
| Naive Bayes          | 0.23             | 0.05         | 94.91%     |
| KNN                  | 1.26             | 29.54        | 98.91%     |
| Logistic Regression Classifier | 35.69          | 0.06         | 98.36%     |
| Random Forest Classifier | 9.46            | 0.05         | 86.91%     |
| SVM                  | 222.75           | 5.89         | 98.91%     |
| Our method           | 40.53            | 0.53         | 99.09%     |
It can be seen from the comparison that our method has the highest recognition accuracy though cost some time on training. Fortunately, training time has little impact on industrial production. As for test time, the speed of identifying 550 characters at 0.53s is industrially acceptable.

6. Conclusion
This paper proposes a method for automotive parts coding character recognition. The algorithm proposed in this paper is not only able to localize the character region accurately in the case of reflection, abrasion, smudge and etc., but also complete the division of every single character without prior knowledge like the exact number of characters. The convolutional neural network constructed in this paper can quickly identify the segmented characters. The experimental results show that the method can complete the code recognition quickly and accurately. Improving the recognition speed and accuracy will be the topic of future work.

Acknowledgments
This work is supported by The National Key Research and Development Program of China(2018YFB1701203); Projects for comprehensive and standardized trial as well as new model application of smart manufacturing (new model application of smart manufacturing for Changan new energy automobile powertrain); Graduate research and innovation foundation of Chongqing, China (CYB18019).

References
[1] Matas J, Chum O, Urban M and Pajdla, T. (2004) Robust wide-baseline stereo from maximally stable extremal regions Image and Vision Computing. 22(10) 761-767
[2] Neumann L, and Matas J. (2016) Real-time lexicon-free scene text localization and recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence. 38(9) 1872-1885
[3] Wu T. (2016) A Degraded Character of Printed Number Recognition Algorithm. International Conference on Intelligent Human-machine Systems & Cybernetics. IEEE.
[4] Afroge S, Ahmed B, and Mahmud F. (2016) Optical character recognition using back propagation neural network. International Conference on Electrical. IEEE.
[5] Jaderberg M, Simonyan K, Vedaldi A and Zisserman. (2016) Reading text in the wild with convolutional neural networks. International Journal of Computer Vision 116(1) 1-20
[6] Tian Z, Huang W, He T and Qiao Y. (2016) Detecting text in natural image with connectionist text proposal network.
[7] Shi B, Bai X, and Yao C. (2016) An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence 39(11) 2298-2304
[8] Nan Yang, Bai Ruilin, Ji Feng. (2015) Online visual detection of canning printed characters. Optical Technique 41(6) 489-493
[9] Pei Mingtao, Wang Yongjie and Jia Yunde. (2014) License Plate Character Segmentation Based on Multiple Scale Templates Matching and Part-Based Model. Transactions of Beijing Institute of Technology, 34(9) 961-965, 971
[10] Xu B, Wang N, Chen T and Li M. (2015) Empirical evaluation of rectified activations in convolutional network. Computer Science
[11] Hinton G E. (2010) Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair International Conference on International Conference on Machine Learning.
[12] Kingma D and Ba J. (2014) Adam: a method for stochastic optimization. Computer Science.
[13] Duchi J, Hazan E, Singer Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12 (Jul) 2121-2159.
[14] Zeiler, Matthew D. (2012) ADADELTA: An Adaptive Learning Rate Method. Computer Science.