An Improved OCR Recognition Algorithm of Electronic Components Based on Self-adaptation of Multifont Template

[*]Tao Lin, [+]Zhuqing Jia
[*]SICE, Beijing University of Posts and Telecommunications, Beijing, 100876, China
Email: ausjtgn@gmail. com
[+]SICE, Beijing University of Posts and Telecommunications, Beijing, 100876, China
Email: justin@jusjustin. com

Abstract: **The recognition method of Optical Character Recognition has been expensively utilized, while it is rare to be employed specifically in recognition of electronic components. This paper suggests a high-effective algorithm on appearance identification of integrated circuit components based on the existing methods of character recognition, and analyze the pros and cons.**

Key words: Optical Character Recognition; Fuzzy Page Identification; Mutual Correlation Matrix; Confidence Self-adaptation

1 Introduction

Optical Character Recognition utilizes the high calculation performance of computers to process and obtain text information from image data efficiently, with the specific applications in diverse fields such as the recognition of cars' licenses or the classification of books in library. In response to demands of intelligent management and classification of electronic components, we propose a series of algorithms based on the existing OCR algorithm and the characteristics of electronic components.

2 Main challenges

The electronic components this paper researches refers to integrated circuit chips, with the major characteristics as below: (1) tiny in size and the quality of image acquired is easily affected by the shooting angle and the light intensity, etc. (2) Different manufacturers employ different fonts as a mark, which decreases the success rate of recognition to a great extent.

Common optical recognition algorithms are not suitable for these characteristics, so we need to make pertinent improvements to enhance the precision of recognition. We have collected various fonts of different chips manufacturers as font templates before the works begin. The algorithm processes are shown below:

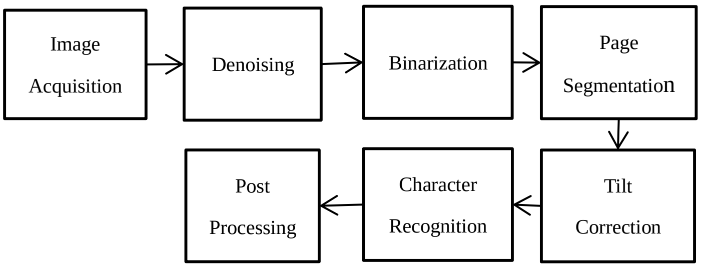


Figure 1

3 Key technologies

3.1 Image acquisition

CMOS cameras are used in image acquisition in this paper, and we choose lens of short focal length to match the size of electronic components. Due to the extensive application, we don't introduce it in details here.

3.2 denoising and binarization

First denoise the color image acquired. Denoising can improve the signal-noise ratio of the image to some extent and enhance the precision of recognition. Methods of denoising include Medium Filtering, Gaussian Filter, Wiener Filtering, Wavelet Denoising, etc. We adopt the classic Gaussian Filter for denoising. [1]

The color image acquired in RGB system needs to be transformed into grayscale image by Luminance Equation:

$$\boldsymbol{Y = 0.299R + 0.587G + 0.114B}$$

Then the grayscale image needs to be binarized into two values by a specific threshold. The selection of threshold is important, and there are two main kinds of thresholds: static thresholds and dynamic thresholds. Although dynamic threshold methods with self-adaptation can improve the details of binarized image, they will always bring unnecessary noise in applications of OCR, such as the pins image in electronic chips. We use OSTU to determine the global static threshold. [2]. We took a surface picture of a certain type of integrated circuit chips, and the binarized image is as below:

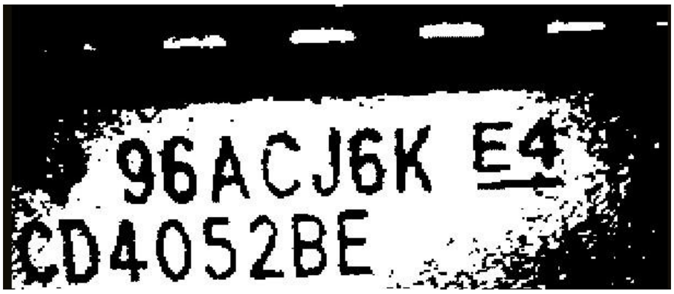


Figure 2

3.3 page segmentation

The normal sequence of recognition process is that first tilt correction then character segmentation, while in this paper we directly segment the characters without tilt correction. This is because: (1) the chip image acquired can't provide obvious straight features to do tilt correction; (2) as long as the inclination of the image is within a specific range, we are able to correct the image precisely by the information based on the page characteristics, which will be introduced later.

In OCR systems, there are two main methods of character segmentation: projection analysis and connected component process. The anti-noise performance of projection analysis is not good enough to handle the rough image obtained above and projection analysis requires tilt correction at first. So we choose the connected component process to segment the page, specifically use the algorithm of connected components labeling.

First define the bounding. Bounding is a set of pixels which satisfy the equation below in the binarized image (where white background is represented by 1, black foreground is represented by 0):

$$\boldsymbol{edge} = \{x_{i,j} | x_{i,j} = 0 \wedge x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1} > 0\}$$

Based on the definition, connected components labeling (method of Region Growing) can classify all the black pixels in the image into separate sets, where the black pixels are tightly adjacent to each other. [3] Then we can obtain a whole set consists of several connected components tagged by different numbers.

In order to select connected components of characters from all connected components, we need to analyze some features of connected components, which are defined as below:

Definition 1 Minimum Bounding Rectangle is the rectangle of a connected component with the minimal area and all connected component pixels contained in it:

$$\boldsymbol{Rect}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{width}, \boldsymbol{height})$$

where $\boldsymbol{x}, \boldsymbol{y}$ are the coordinate of vertex on the top left, $\boldsymbol{width}, \boldsymbol{height}$ are of the rectangle.

Definition 2 $\boldsymbol{x_m}, \boldsymbol{y_m}$ are the middle point coordinate of a connected component:

$$\boldsymbol{x_m} = \boldsymbol{x} + \frac{\boldsymbol{1}}{\boldsymbol{2}} \times \boldsymbol{width}$$

$$\boldsymbol{y_m} = \boldsymbol{y} + \frac{\boldsymbol{1}}{\boldsymbol{2}} \times \boldsymbol{height}$$

Definition 3 $\boldsymbol{S}$ is the area of a connected component, namely the number of pixels that make up the whole connected component.

Definition 4 The ratio of a connected component is defined as:

$$\boldsymbol{R_{wh}} = \boldsymbol{width}/\boldsymbol{height}$$

Definition 5 The area of the Minimum Bounding Rectangle of a connected component is defined as:

$$\boldsymbol{S_r} = \boldsymbol{width} \times \boldsymbol{height}$$

Definition 6 The compactness of a connected component is defined as:

$$\boldsymbol{R_{com}} = \boldsymbol{S}/\boldsymbol{S_r}$$

According to the definitions above, we can identify the character connected components by setting specific standards. After a large amount of experiments, we choose （$\boldsymbol{S_r} > 20 \wedge 0.3 < \boldsymbol{R_{wh}} < 1 \wedge \boldsymbol{R_{com}} < 0.7$）, and obtain a set of character connected components $\{\boldsymbol{c_i}\}$.

It has been found that the precision of connected components labeling is high in cases that illumination is adequate and no character fracture exists. Whereas, in cases of inadequate illumination and character fracture, the algorithm above can't obtain complete and accurate information of characters. For example, the result of connected components labeling and basic page segmentation on the electronic chip surface image above is shown as figure 3:

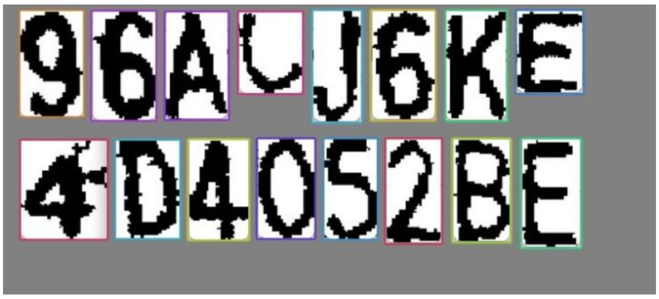


Figure 3

In this result, most character connected components have been found, while the top half of the fourth character “C” has been discarded due to character fracture.

Therefore, we propose Fuzzy Page Segmentation to improve the performance of segmentation algorithm, which implements fuzzy reprocessing on the page information obtained through the basic page segmentation. Fuzzy Page Segmentation consists of two parts: reconnection of fractured characters and completion of lost characters.

3.3.1 Reconnection of fractured characters

Find and reconnect the discarded parts of characters because of character fracture to the original characters. Specific algorithm is described as below:

(1) Find out the minimal $\boldsymbol{y_i}$ and the maximal $\boldsymbol{height_i} + \boldsymbol{y_i}$ of character connected components in each row. Reconstruct a set of fuzzy rectangles $\{\boldsymbol{r_i}\}$, and each $\boldsymbol{r_i}$ satisfies that (a) the coordinate of vertex on the top left is $(\frac{1}{2}(\boldsymbol{x_i} + \boldsymbol{x_{i-1}}), \min(\boldsymbol{y_i}))$; (b) the width of each $\boldsymbol{r_i}$ is $\boldsymbol{width_i} + \frac{1}{2}(2\boldsymbol{x_i} + \boldsymbol{x_{i+i}} + \boldsymbol{x_{i-1}})$; (c) the height is $\max(\boldsymbol{height_i} + \boldsymbol{y_i}) - \min(\boldsymbol{y_i})$.

(2) Traverse each discarded connected components $\boldsymbol{c_j}$ after basic segmentation. If the area ratio of $\boldsymbol{c_j}$ to $\boldsymbol{r_i}$ reaches a certain threshold (0.1 in this paper), then determine $\boldsymbol{c_j}$ is one part of a character, and reconnect it to the corresponding character according to its coordinate of vertex on the top left.

Set of fuzzy rectangles is shown as follow:

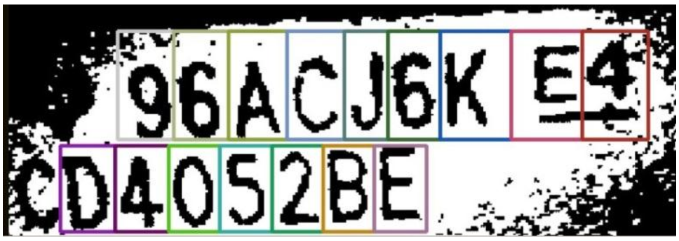


Figure 4

2. Completion of lost characters

Find the potential lost characters because of the adhesion to background. In order to demonstrate this algorithm, we intentionally link character “6” to background to be erased as below.

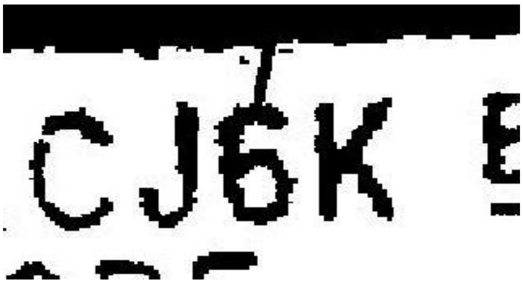


Figure 5

Completion of lost characters algorithm is aimed at this situation, and the specific description is as follow:

(1) Find out the minimal $\boldsymbol{y_i}$ and the maximal $\boldsymbol{height_i} + \boldsymbol{y_i}$ of character connected components in each row. Reconstruct a set of slit rectangles $\{\boldsymbol{s_i}\}$, and each $\boldsymbol{s_i}$ satisfies that (a) the coordinate of vertex on the top left is $(\boldsymbol{x_i} + \boldsymbol{width_i}, \min(\boldsymbol{y_i}))$; (b) the width of each $_{\boldsymbol{i}}$ is $(\boldsymbol{x_{i+1}} - \boldsymbol{width_i} - \boldsymbol{x_i})$; (c) the height is $\max(\boldsymbol{height_i} + \boldsymbol{y_i}) - \min(\boldsymbol{y_i})$.

(2) Find the secondary connected components in each $\boldsymbol{s_i}$. If the area ratio of the Minimum Bounding Rectangle to $\boldsymbol{s_i}$ reaches a certain threshold (0.3 in this paper), then determine here exists a character. Reuse connected components labeling on this $\boldsymbol{s_i}$ to find the lost character connected component, and adjust it into normal size based on the average information of other character connected components (such as coordinate of $\boldsymbol{x}, \boldsymbol{y}$).

The slit rectangles reconstructed is shown as follow:

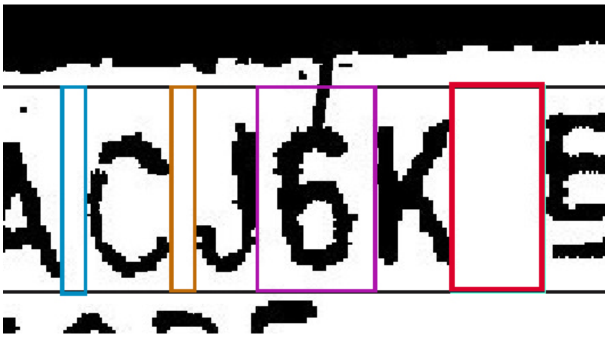


Figure 6

It can be seen that Fuzzy Page Segmentation is able to handle conditions of character fracture and character loss well.

3.4 Tilt correction based on the page information

Precise tilt correction can be carried out through page information obtained. Implement Linear Regression on middle points of character connected components in each row, to acquire slopes of every row. Choose the one with the largest coefficient of correlation to be the slope of all character connected components, and compute its arc tangent which will be used in the rotation formula to revise each character connected component:

$$x' = (x - x_m) * cos(Angle) + (y - y_m) * sin(Angle) + x_m$$
$$y' = (y - y_m) * cos(Angle) + (x - x_m) * sin(Angle) + y_m$$

where $\boldsymbol{angle}$ is arc tangent value of the slope, $\boldsymbol{x_m, y_m}$ is middle point of each character connected component, $\boldsymbol{x, y}$ is pixels in character connected components.

3.5 Character recognition

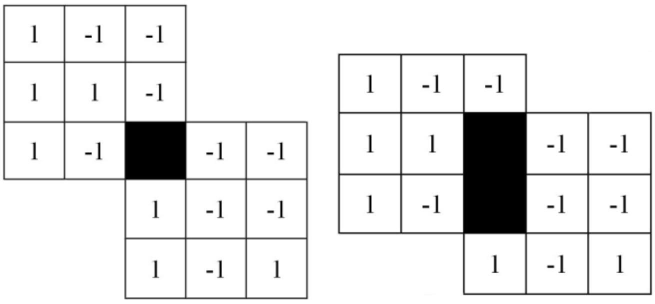
Algorithms of character recognition are mainly divided into template feature matching algorithms based on structure or statistics. Beacuse various fonts of different manufacturers might have the same statistical feature, algorithms of statistics will decrease the matching precision if used in electronic chips recognition. We choose the structural matching algorithm and select 20 kinds of fonts of several main chips manufacturers as font templates. The specific recognition process in this paper is first to compute coefficient of correlation between matrixes to match and templates, then to find out the most matching template.

3.5.1 Definition of coefficient of correlation

For the convenience of coefficient calculation, replace all 0 pixels with -1. Give the definition of two-dimensional Mutual Correlation Matrix: $\boldsymbol{D = A \otimes B}$, where $\boldsymbol{D}$ is Mutual Correlation Matrix, $\boldsymbol{A}$ is input (the matrix to match), $\boldsymbol{B}$ is convolution kernel(the template matrix), $\otimes$ represents convolution between matrixes. Each element $\boldsymbol{d_{i,j}}$ in $\boldsymbol{D}$ represents the sum of corresponding elements' product in the overlapped area of $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\boldsymbol{d_{i,j}} = \sum_{\substack{\boldsymbol{i,j} \in \\ \text{overlapped} \\ \text{area}}} \boldsymbol{b_{i,j} \times a_{m-i+1,n-j+1}}$$

where $0 < \boldsymbol{i} < 2\mathrm{m} - 1, 0 < \boldsymbol{j} < 2\mathrm{n} - 1$. It is apparent that $\boldsymbol{D}$ is a matrix of (2m-1) × (2n-1).



We propose two ways computing coefficient of correlation: matched filter and mutual correlation filter. Both of them use template matrixes as the kernel of filter, while the outputs we need are different.

(1) Method of matched filter

Matched filter utilizes the center element of mutual correlation matrix as output (coefficient of correlation). According to the matched filter in telecommunication system, the center element represents the situation of the matrix to match that overlaps completely with the convolution kernel. Its value is exactly the output of matched filter, namely the sum of product of each corresponding point in matrix to match and template matrix. Define the normalized coefficient of correlation of matched filter as:

$$\boldsymbol{r_1} = \frac{\boldsymbol{d_{x_m, y_m}}}{\sum \boldsymbol{b_{m,n}}^2}$$

where $\boldsymbol{x_m, y_m}$ is coordinate of center element in the mutual correlation matrix, $\boldsymbol{b_{m,n}}$ is any element in convolution kernel, denominator in the formula is also the number of elements in convolution kernel. Normalized coefficient is convenient for comparison.

Matched filter utilizes only one element of mutual correlation matrix, so calculation of it can be simplified to reduce time complexity. While only a little information of mutual correlation matrix has been used, and it is too sensitive to translation deviation. Therefore matched filter method is less effective than mutual correlation filter.

(2) Method of mutual correlation filter

Mutual correlation filter utilizes the largest value of the mutual correlation matrix as output. In fact, the output value of matched filter is often not the largest, because the image will be still deviated and translated to some extent after page segmentation and tilt correction. Method of mutual correlation can perfectly handle this problem.

Mutual correlation method takes deviation distance into account. For example, when character "E" and "I" being matched with convolution kernel "E", the largest value is the same but their location are different ("E" is at the middle left of the mutual correlation matrix, while "I" is at the center). So we make a punishment on the largest value which does not appear at the center of convolution kernel matrix (In this paper subtract Euclidean distance between the largest point and the center point). Therefore, the adjusted $\boldsymbol{d_{max}}'$ is:

$$\boldsymbol{d_{max}}' = \begin{cases} \max(\boldsymbol{d_{i,j}}) \quad , & \max(\boldsymbol{d_{i,j}}) \text{ at the center} \\ \max(\boldsymbol{d_{i,j}}) - \sqrt{(\boldsymbol{i} - \boldsymbol{x_m})^2 + (\boldsymbol{j} - \boldsymbol{y_m})^2}, & \text{otherwise} \end{cases}$$

To be compared with matched filter, the coefficient of correlation of mutual correlation method is normalized as follow:

$$\boldsymbol{r_2} = \frac{\boldsymbol{d_{max}}'}{\sum \boldsymbol{b_{m,n}}^2}$$

3.5.2 Character matching

Before the character matching, fill all the character connected components acquired in order to implement character matching based on character templates. Also we should use bilinear interpolation [4] to scale the input into apposite size. According to the variety of fonts used by different chips manufacturers, we propose the self-adaptive matching algorithm of confidence.

Using the correct font of templates to match can increase the accuracy largely due to the variety of fonts of different manufacturers. Normal matching algorithms usually compare input with character templates of all fonts and doesn’t exploit the latent information that all characters on a chip surface is of the same font. This will result in unnecessary calculation, and these algorithms are hardly viable if there are plenty kinds of fonts. Therefore, we design a self-adaptive matching algorithm of confidence that can adjust the font of character templates automatically during matching.

Define $\boldsymbol{r_{con}}$, the confidence of a font towards the input matrix, as the largest coefficient of correlation among all the character templates of this font to the input matrix (the matrixes to match is different character templates of the same font, and the convolution kernel is the input matrix). Additionally, we prescribe that matched filter method is firstly used, and when reasonable outcome is inaccessible, mutual correlation method is subsequently used. Namely, first calculate all $\boldsymbol{r_{(1)}}$ and obtain the $\boldsymbol{r_{con(1)}}$. If $\boldsymbol{r_{con(1)}} \geq \mathrm{mean}\left(\boldsymbol{r_{(1)}}\right) + 3 \times \mathrm{std}\left(\boldsymbol{r_{(1)}}\right)$, determine that this coefficient is suitable. Or change into mutual correlation method and recalculate $\boldsymbol{r_{(2)}}$ and $\boldsymbol{r_{con(2)}}$. Mean() and std() above are to obtain the mean and the standard deviation.

According to definitions above, the matching algorithm of confidence self-adaptation is described as follow:

Step 1 Rearrange the image characters and make the characters which are close to image center to be matched firstly. This is to increase the reliability of algorithm.

Step 2 Take one character to be matched with $\boldsymbol{N}$ kinds of fonts one by one, and calculate $\boldsymbol{r_{con}}$ of each font. Meanwhile, the consequence of character recognition is obtained.

Step 3 Arrange all fonts by $\boldsymbol{r_{con}}$ from largest to smallest.

Step 4 Reduce $\boldsymbol{N}$ and repeat Step 2 to obtain the font with the largest average $\boldsymbol{r_{con}}$ and acquire the final outcome of character recognition process.

During the procedures above, each $\boldsymbol{r_{con}}$ of every time iteration is synthesized with previous outcomes to acquire the statistical average. Recognition of character and font are carried out in the meantime. Different convergent methods of $\boldsymbol{N}$ will influence the precision of recognition. Here we use this array to reduce $\boldsymbol{N}$: {0, 0, 0, 1, 2, 4, 8, 16…}, where the elements represent the amount of diminishment of $\boldsymbol{N}$, and the number of 0 is protection length. After a few times of iteration, the correct font will be obtained, and the precision of recognition of the rest characters will be increased largely. It has been proved that this algorithm can improve precision and reduce the amount of calculation.

3.6 Post Processing

Utilize the information of row and column to rearrange the characters by the standard as follow: when the difference of $\boldsymbol{y_m}$ of two character connected components is less than the average height of connected components, determine they are in the same row. In one row components with less $\boldsymbol{x_m}$ are arranged in front, and the rows with less average $\boldsymbol{y_m}$ are arranged in front.

Generally, the final recognition outcome is not exactly correct. For example in this paper, the final outcome is “9SACJSKE4↙D4O52BE”. Namely character “C” on the bottom left is unable to be recognized and “6” has been misrecognized as “S”. Consider that the types of electronic chips to be matched is finite in practice, so we can set up a database with complete surface information of all chips, and compare with the recognition outcome to choose the most likely one as the result.

Here we use Levenshtein algorithm [5] described as follow: define $\boldsymbol{ld}$ as the edit distance of two characters which is the smallest times editing one character to another (here editing represents delete, add and replace). Define the similarity $\boldsymbol{Re}$ between character string $\boldsymbol{A}$ of recognition outcome and string $\boldsymbol{B}$ of chip surface type in the database as follow:

$$\boldsymbol{Re} = \boldsymbol{1} - \frac{\boldsymbol{ld}}{\boldsymbol{max}(\boldsymbol{length}(\boldsymbol{A}), \boldsymbol{length}(\boldsymbol{B}))}$$

Every string $\boldsymbol{B}$ in database has an $\boldsymbol{Re}$. If max($\boldsymbol{Re}$)⩾mean($\boldsymbol{Re}$)+std($\boldsymbol{Re}$), return the chip type corresponding to max($\boldsymbol{Re}$), or determine the recognition outcome is abnormal and require clearer image of chip surface. In this paper the correct outcome of recognition is “96ACJ6KE4↙CD4052BE”.

4 Algorithm analysis

This paper puts forward some innovation and improvements on existing OCR algorithm and proposes recognition algorithm suitable for identification of integrated circuit chips surface type, which largely increases the precision and efficiency of recognition. Main improvements are as follow:

(1) Propose Fuzzy Page Segmentation algorithm to improve page segmentation which can reconnect fractured characters and complete lost characters.

(2) Propose a standard of similarity in character recognition based on coefficient of correlation and mutual correlation matrix.

(3) According to the information of multifonts, propose the self-adaptive matching algorithm of confidence.

Due to the intrinsic defect of fuzzy recognition algorithm, although in most cases the fractured characters can be reconnected and lost characters can be completed, characters largely linked to background and locate at the head or the rear of a row is hardly recognized, such as “C” on the bottom left.

Also there are improvements can be done to algorithms this paper put forward. For example based on the OCR algorithm of neural networks recently proposed, the precision of recognition can be increased. Combined with the large amount of information in database, a new self-learning algorithm of chips surface recognition can be researched and developed later.

5 Reference

[1] Ito, K. Gaussian filter for nonlinear filtering problems[J]. Proceedings of the IEEE Conference on Decision and Control, 2000, (2): 1218-1223.

[2] Lin, Guo-Yi; Zwang, Wei-Gang. Image segmentation of the Ostu method based on EP algorithm[J]. Chinese Journal of Sensors and Actuators, 2006, (17): 179-182.

[3] He, Lifeng; Zhao, Xiao; Chao, Yuyan. Configuration-Transition-Based Connected-Component Labeling[J]. IEEE TRANSACTIONS ON IMAGE PROCESSING, 2014, 23(2): 943-951.

[4] Lu, Jing; Xiong, Si; Wu, Shihong. An improved bilinear interpolation algorithm of converting standard-definition television images to high-definition television images[J]. 2009 WASE International Conference on Information Engineering, ICIE 2009, 2009, (2): 441-444.

[5] Xu, Jianhua; Zhang, Xuegong. Kernels based on weighted levenshtein distance[J]. IEEE International Conference on Neural Networks - Conference Proceedings, 2004, (4): 3015-3018.