Journal of Electronic Research and Application, 2024, Volume 8, Issue 4
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Chip Surface Character Recognition Based on OpenCV

Lihang Yin*

Henan Open University, Zhengzhou 450008, China

***Corresponding author:** Lihang Yin, yinlihang2018@163.com

Abstract: Chip surface character recognition is an important part of quality inspection in the field of microelectronics manufacturing. By recognizing the character information on the chip, automated production, quality control, and data collection and analysis can be achieved. This article studies a chip surface character recognition method based on the OpenCV vision library. Firstly, the obtained chip images are preprocessed. Secondly, the template matching method is used to locate the chip position. In addition, the surface characters on the chip are individually segmented, and each character image is extracted separately. Finally, a Support Vector Machine (SVM) is used to classify and recognize characters. The results show that this method can accurately recognize the surface characters of chips and meet the requirements of chip quality inspection.

Keywords: Template matching; Character recognition; SVM; OpenCV

Online publication: August 20, 2024

1. Introduction

The quality inspection of chip surface characters plays an important role in reducing manufacturing defects and improving the quality and reliability of chip products. In chip surface character recognition, commonly used methods include template matching, feature extraction, deep learning, and statistical model-based methods. The template matching method determines the best matching character by matching it with predefined character templates [1,2]. This method is suitable for situations where the character shape is regular and the background interference is small. It performs character recognition by comparing the similarity with the template. The method based on feature extraction usually extracts the shape, texture, edge, and other features of characters, establishes feature vectors, and uses classifiers for character classification and recognition [3-6]. The method based on deep learning utilizes deep learning models such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) to learn and extract features from a large amount of data, thereby achieving recognition of chip characters [7-11]. Statistical model-based methods typically use Hidden Markov Models, Conditional Random Fields, and other models to model and recognize character sequences [12,13]. This article uses C++ programming language and the computer vision library OpenCV to achieve recognition of surface characters on chips [14].

2. Chip vision system

Chip machine vision systems typically consist of computers, motion control modules, camera modules, etc. The images of the chips come from the production line, and the industrial camera model is MV-CU120-10GC, with a color resolution of 4,024 × 3,036. Chip surface character recognition includes image acquisition, chip image preprocessing, chip positioning, character region acquisition, character segmentation, and character recognition, as shown in **Figure 1**.

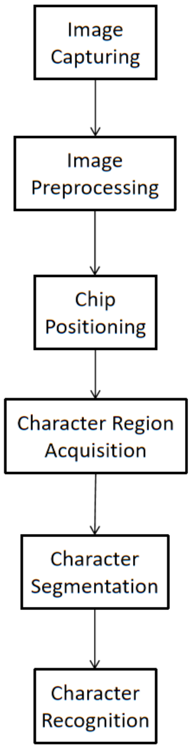


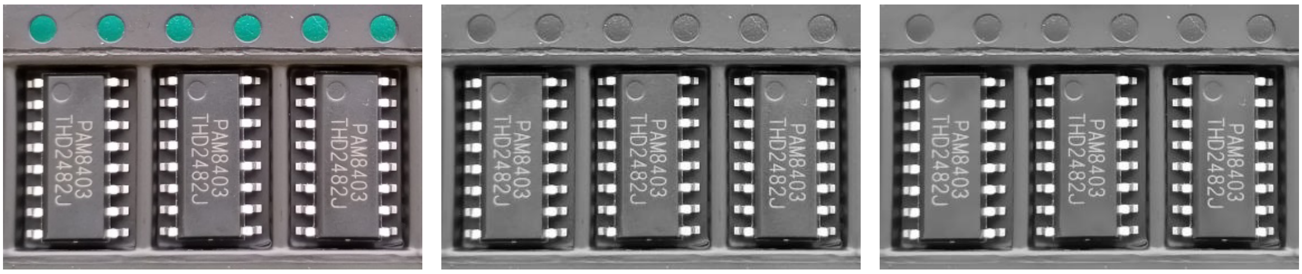
Figure 1. Chip surface character recognition process

3. Chip image preprocessing

The purpose of chip image preprocessing is to reduce noise in the image, highlight character features, and thereby improve recognition accuracy and robustness. Chip image preprocessing includes image grayscale and image filtering. Image grayscale is the process of converting color images into grayscale images. In character recognition, only the brightness information of characters is usually considered. Grayscale can convert the red, green and blue (RGB) channel values of each pixel in a color image into a single grayscale value. This can not only reduce the dimensionality of the data and simplify the complexity of subsequent processing, but also better highlight the brightness differences of characters. For the chip image, the average weighting method is used to convert the image into a grayscale image, and the weighted values of the RGB three channels are 0.299, 0.587, and 0.114, respectively.

Due to the rich details of the chip image, bilateral filtering is used to enhance the chip image. Bilateral filtering is a non-linear filtering method, and its filter kernel consists of two functions: spatial domain kernel and value domain kernel. It can not only remove noise but also protect image details. OpenCV provides the bilateral filter function to implement bilateral filtering operations. The specific method is: void cv::bilateralFilter(InputArray src,OutputArray dst,int d,double sigmaColor,double sigmaSpace,int borderType = BORDER_DEFAULT).

Figure 2 shows a comparison between converting the original image of the chip to grayscale and using bilateral filtering.



(a) Original image (b) Color to grayscale image (c) Bilateral filtered image

Figure 2. Chip image preprocessing

4. Chip positioning

Before performing character segmentation, it is necessary to locate the position of the chip in order to determine the position of the chip characters and improve detection speed. A method based on normalized cross-correlation is used to locate the chip. The normalized cross-correlation matching function in OpenCV is: void matchTemplate (InputArray image, InputArray template, OutputArray result, int method, InputArray mask=noArray()), where method=cv:: TM_CCOEFF_NORMED.**Figure 3** shows the effect of using this method for chip positioning.

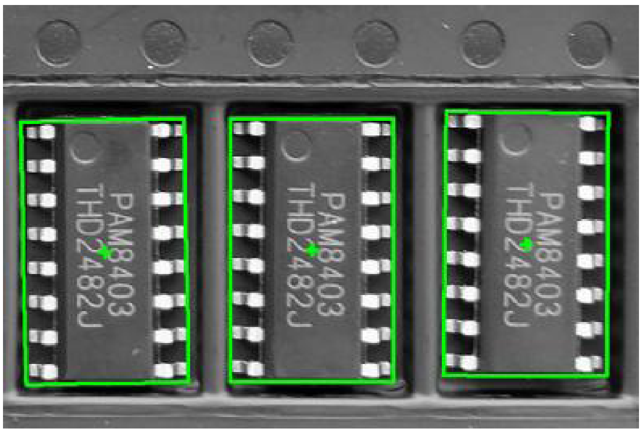


Figure 3. Chip positioning results

5. Chip surface character segmentation

5.1. Binarization of chip images

In character recognition, we usually only focus on the shape and edge information of characters. Here, we use binarization technology to process chip images. Binarization is the process of converting a grayscale image into a black and white binary image. Through binarization, the pixel values in the grayscale image can be segmented according to a threshold. Pixels with brightness higher than the threshold are set to white (255), and pixels with brightness lower than the threshold are set to black (0). This can better highlight the edges and contours of characters, making it easier for subsequent character segmentation and recognition.

5.2. Morphological filtering

Morphological filtering is a series of shape-based image processing operations commonly used in Digital Image Processing [15]. Since there will be slight fractures on the characters on the chip surface, we can use the closing operation to deal with it. The calculation process of closed operations is to first expand and then corrode. It can connect adjacent pixels, fill small holes in the target, and smooth the boundaries without significantly changing the area of the region. The function of morphological filtering in OpenCV is: void morphologyEx (InputArray src, OutputArray dst, int op, InputArray kernel, Point anchor=Point (-1, -1), int iterations=1, int borderType=BORDER_CONSTENT, const Scalar&borderValue=morphologyDefault BorderValue). When using a closed operation, the parameter op is MORPH_CLOSE.

5.3. Character segmentation

In chip surface character recognition, characters are usually arranged consecutively together with small intervals between them. To accurately recognize each character, it is necessary to segment the character from the image and use it as a separate image block for training and feature acquisition. This system adopts a contour based character segmentation method, using the functions provided by the OpenCV image processing library, findContours and boundingRect. These two functions can find the character contours in the image and determine the bounding boxes of each character. Their definitions are as follows:

```
findContours(binaryImage,contours,RETR_EXTERNAL,CHAIN_APPROX_SIMPLE);
boundingRect (InputArray array);
```

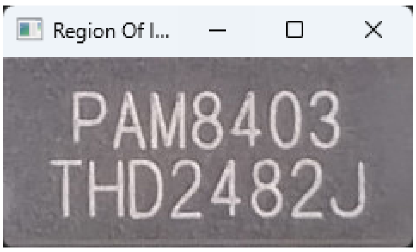
The findContours function takes a binary image as input and returns a vector containing all contours. A contour is a closed curve composed of a series of points that surround the edges of a character. By traversing the contour vector, the contour information of each character can be obtained. The boundingRect function takes the contour as input and returns a rectangular box that completely surrounds the contour. By calculating the bounding boxes of each contour, the position and size of each character can be obtained. However, not all contours represent characters, so when performing character segmentation, it is necessary to add some filtering conditions to remove inappropriate bounding boxes. Filtering is performed by setting the minimum width, minimum height, and aspect ratio to select the character bounding boxes that meet the conditions, and ultimately obtain the character segmentation result.The core code for character segmentation is as follows:

```
vector<vector<Point>> contours; vector<cv::Rect> charRects; Mat colorImage;
findContours(closing, contours, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE);
int minWidth = 10; int minHeight = 10; float aspectRatioMin = 0.2; float aspectRatioMax =2;
cvtColor(closing, colorImage, COLOR_GRAY2BGR);
for (const auto& contour : contours)
 {
    Rect rect = cv::boundingRect(contour);
    if (rect.width >= minWidth && rect.height >= minHeight)
    {
```

```
            float aspectRatio = static_cast<float>(rect.width) / rect.height;
            if (aspectRatio >= aspectRatioMin && aspectRatio <= aspectRatioMax)
            {
                charRects.push_back(rect);
                rectangle(image, rect, cv::Scalar(0, 255, 0), 2);
                rectangle(colorImage, rect, cv::Scalar(0, 255, 0), 2);
            }
        }
    }
```

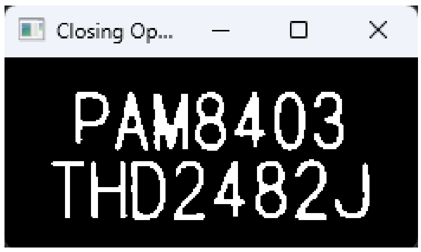
6. Surface character recognition

Through the previous processing, images containing individual characters have been obtained, and then these character images can be classified and recognized. A Support Vector Machine (SVM) is used to classify the surface characters of the chip. Support Vector Machine (SVM) is a supervised learning algorithm widely used in the fields of pattern recognition, machine learning, and data mining. The core idea of a support vector machine is an optimal decision hyperplane that maximizes the interval between classification boundaries and sample points, effectively separating sample points of different categories. SVM can effectively handle nonlinear problems, is suitable for high-dimensional spatial data classification, and the model is relatively simple and easy to implement. In addition, the choice of SVM kernel function is also very important. The kernels of SVM include linear kernel function (linear), polynomial kernel function (poly), radial basis kernel function (RBF), and sigmoid kernel function. Through testing, when the kernel function is selected as "RBF", the recognition rate is highest, so the radial basis function is chosen here. This experiment was conducted under the conditions of Inter (R) Core (TM) i5-7500 CPU @ 3.40 GHz, 16 GB memory, and SSD256G. **Figure 4** shows the recognition results of the chip's surface characters.

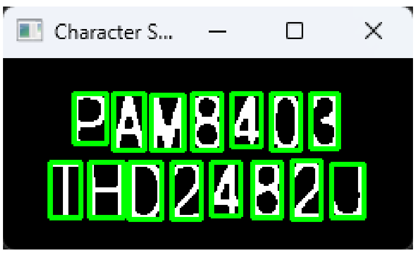


(a) Get character region

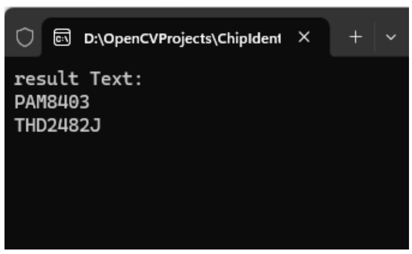
(b) Binarization



(c) Closing operation



(d) Character segmentation



(e) Recognition results

Figure 4. Chip character recognition

7. Conclusion

This article focuses on the recognition of printed characters on the surface of chips and implements an efficient and accurate recognition method using C++ language combined with the OpenCV library. After testing, it can meet the needs of detection and provide reliable assurance for quality control in the field of microelectronics manufacturing. However, due to the rich font styles and inconsistent character sizes on the chip surface, there is no unified standard. To further improve the robustness of character recognition, the next step will be to train a large number of custom character models to adapt to more character sets and application scenarios.

Funding

Henan Province Science and Technology Research Project "Key Technologies for Intelligent Recognition of Chip Surface Defects Based on Machine Vision" (Project No. 242102210161).

Disclosure statement

The author declares no conflict of interest.

References

[1] Zhao J, Chen W, Li G, et al., 2022, Research on Template Matching Character Wheel Recognition Method Based on Canny Operator. Instrumentation Technology, 2022(01): 48–52.

[2] Wang S, Liu G, Li J, et al., 2024, Research on License Plate Character Recognition Algorithm Based on Template Matching. Journal of Wuhan Institute of Shipbuilding Technology, 23(01): 97–102.

[3] Hu L, Sun H, Fan A, 2013, Research on Feature Extraction in Handwritten Character Recognition. Automation & Instrumentation, 2013(02): 15–17.

[4] Huang F, Li X, Wang K, et al., 2023, Diode Character Recognition Based on Grayscale Co-Occurrence Matrix Texture Features and MLP. Journal of Jiangsu University of Technology, 29(02): 64–71.

[5] Yang M, Chen L, Zheng B, 2012, Character Recognition Feature Extraction Based on Lifting Wavelet. Journal of Jilin Institute of Chemical Technology, 29(11): 97–99.

[6] Zhu S, 2014, Research on License Plate Recognition Algorithm Based on Character Multi-Feature Extraction, thesis, Central South University.

[7] Yu Z, 2023, Research on Circuit Board Character Detection Method Based on Deep Learning, thesis, Jiangnan University.

[8] Weng L, Liu L, 2023, Application of Machine Vision Based on Deep Learning in Industrial Instrument Inspection. Intelligent Manufacturing, 2023(05): 40–44.

[9] Zhang C, 2016, Character Recognition Based on Deep Learning, thesis, University of Electronic Science and Technology of China.

[10] Li J, 2023, Research on Inkjet Character Recognition and Verification Based on Deep Learning, thesis, Fujian University of Technology.

[11] Wang R, Ma W, 2021, Character Recognition Algorithm Based on BP Network. Modern Computer, 27(24): 47–50.

[12] Wang X, 2013, Offline Handwritten Chinese Character Recognition Based on BP Neural Network and HMM Hybrid Model, thesis, Tianjin Normal University.

[13] Yang Y, 2013, Research on Image Classification Method Based on Conditional Random Fields, thesis, East China

Normal University.
[14] Mao X, Leng X, Deng, 2015, Introduction to OpenCV3 Programming. Electronic Industry Press, Beijing.
[15] Gonzalez RC, Woods RE, 2020, Digital Image Processing (Fourth Edition). Pearson, London.