Component Pin Recognition Using Algorithms Based on Machine Learning

Yang Xiao¹, Hong Hu¹ *, Ze Liu¹ and Jiangchang Xu¹
¹Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, Guangdong 518055, China

*Email: honghu@hit.edu.cn

Abstract. The purpose of machine vision for a plug-in machine is to improve the machine's stability and accuracy, and recognition of the component pin is an important part of the vision. This paper focuses on component pin recognition using three different techniques. The first technique involves traditional image processing using the core algorithm for binary large object (BLOB) analysis. The second technique uses the histogram of oriented gradients (HOG), to experimentally compare the effect of the support vector machine (SVM) and the adaptive boosting machine (AdaBoost) learning meta-algorithm classifiers. The third technique is the use of an in-depth learning method known as convolution neural network (CNN), which involves identifying the pin by comparing a sample to its training. The main purpose of the research presented in this paper is to increase the knowledge of learning methods used in the plug-in machine industry in order to achieve better results.

1. Introduction

With the development of manufacturing, the demand for industrial automation has increased exponentially. The electronic manufacturing industry is a large-scale production industry that continuously produces automated equipment that is in high demand.

As an important part of the electronics manufacturing industry's automated production line, the plug-in machine is a number of regular electronic components automatically and standardly inserted into the printed circuit board of an item of mechanical equipment.

The basic control process of the plug-in machine is as follows: shoot a mark point into the printed circuit board, prepare the components, qualify with camera detection, and position the plug-in. The role of machine vision in the plug-in machine is mainly to improve the accuracy of the plug-in [1]. Machine vision can be divided into two parts: the first part is getting the mark coordinates and the second part is waiting for the device to be qualified to determine the location of the components and angle of deviation. This paper focuses on the second part of the machine vision work.

Figure 1. Component renderings
Figure 1 shows the effect of the device to be inserted. The main purpose of the algorithm is to identify the component pins. The core algorithm of traditional image processing is binary large object (BLOB) analysis, which has a high recognition rate and a good effect. However, when light conditions are not ideal, the effect will decline. In addition, when dealing with male/female connector components, a template-matching algorithm is needed to realize the effect. That is, the algorithm needs to be specifically for the corresponding component.

In the emergence of artificial intelligence, machine learning is undoubtedly the most amazing aspect. In this paper, we first extract the histogram of oriented gradient (HOG) characteristics of the samples, and then use the support vector machine (SVM) and the adaptive boosting machine (AdaBoost) learning meta-algorithms for classification learning. However, the merits of the machine learning effect depends heavily on the choice of features, such that if the feature selection is wrong it is likely that the test error will be too large. To some extent, convolution neural network (CNN) can overcome this traditional machine learning shortcoming, but CNN construction requires a lot of experience and getting good test results is not easy.

2. Image processing

For recognition of pin components, the traditional image processing is divided into three steps. First, the region of interest (ROI) is extracted from the acquired image. Then, the binarization of the image is performed. Finally, BLOB analysis is performed to obtain the pin area and the center of gravity position.

2.1 Image binarization

Figure 2 shows the extracted ROI area.

![Figure 2. ROI area image](image)

There are basically two binarization methods: the global threshold method or the local threshold method [2]. Because the analysis is highly dependent on binarization, it is a very important step when using tools.

Because of the high contrast between the pin image and the background, the global threshold method can be used. In this paper, the maximum interclass variance method (OTSU) is used. This method divides the image into two parts by the principle of minimizing the rate of misclassification, and then determining the segmentation threshold for the purpose of optimizing the weighted sum of the variance in the class. Figure 3 shows the effect of binarization.

![Figure 3. Binary renderings](image)

2.2 BLOB analysis

BLOB analysis refers to blocks that have similar image features such as texture and color, and spatially connected pixels. The BLOB analysis tool separates the foreground and background of the image, and then gives morphological parameters of any foreground shape; it can also detect round objects. In this process, the BLOB does not parse a single pixel but rather analyzes the rows of BLOB
blocks. This approach can be faster than the pixel-based algorithm. It provides the number, location, area, perimeter, shape and orientation of the spots in the image for the machine vision application [3].

An important step in the BLOB analysis is determination of the connected area. The different analyzes according to the processing method are basically divided into two types: the pixel scanning-based processing method, and the run-length coding method. This article uses a process based on run-length coding whereby the connected area can be extracted to obtain the chain code table and the linear table for each connected area. The chain table can be used to calculate the length and perimeter of the boundary of the region. The linear table can calculate the parameters such as area and center of gravity. Figure 4 shows the results for the BLOB analysis.

![Figure 4. BLOB analysis results](image)

3. Machine learning method

The traditional image processing method is largely determined by image quality problems such as lighting. This article therefore considers using a relatively intelligent algorithm to deal with the problems. Feature extraction of the sample is the most important aspect of machine learning, and so this paper uses the HOG feature with the SVM and AdaBoost algorithm classification. Figure 5 (a) and Figure 5 (b) are positive samples while Figure 5 (c) and Figure 5 (d) are negative samples. To maintain a positive and negative sample balance, 5,800 positive samples and 6,200 negative samples were collected.

![Figure 5. Positive and negative samples](image)

3.1 Extraction of sample HOG characteristics

The gradient direction histogram is simply referred to as the HOG feature and is attributed to Navneet Dalal and Bill Triggs [4]. The HOG feature is also an image feature that is used in image recognition and machine vision, with the most successful application being for pedestrian detection. The HOG feature is concerned with the change of the edge region and the edge shape of the target in the image. The gradient histogram is constructed according to pixels on the edge of the target, and then the histogram features are combined in a certain way for target detection. The specific implementation steps are as follows:

1) Image normalization.
2) Calculate the pixel gradient for each point.
3) Calculate the gradient histogram for each cell.
4) Block the gradient histogram and normalize it.
5) Combine all block gradient histograms to form HOG characteristics.

The HOG characteristics of the positive and negative images extracted from the pair of samples are shown in Figure 6.
3.2 Support vector machines

The support vector machine (SVM) is a two-class model proposed by Corinna Cortes and Vladimir Vapnik [5]. The basic model is defined as the linear classifier with the largest interval in the feature space. The learning strategy is interval maximization, which can be transformed into a convex quadratic programming problem.

The SVM classifies target and non-target objects to establish a separation between them. In general, when solving two kinds of target classification and target linear separability in two-dimensional space, the segmentation criterion is to construct an optimal classification line between the target and the non-target so that one side of the line is all target and the other side is all non-target. However, in reality there is often more than two types of target and there are often linear separability problems. The solution to this lies in the choice of multi-classification and spatial dimension transformation.

The classification process is discussed here under two categories. Assume that the general linear equation is:

$$ g(x) = w \cdot x + b $$

The hyperplane equation is:

$$ w \cdot x + b = 0 $$

The optimal equation for the optimal hyperplane discrimination is:

$$ y_i[(\omega \cdot x + b)] \geq 1, i = 1,2,\ldots,n $$

Satisfying the minimum of $\| \omega \|^2$, the optimal hyperplane can be obtained. In this way, it is transformed into a convex quadratic programming problem, and the Lagrangian duality is transformed into the optimization problem of the dual variable. Then the minimal optimization SMO algorithm is used to solve the dual problem.

By training the parameters $w$ and $b$, the test set can be used to calculate the accuracy rate.

3.3 AdaBoost algorithm

Freund and Schapire when seeking a solution to specific application problems, devised a self-regulating boosting algorithm, which later became known as the Adaboost algorithm [6]. The main principle is that before the training of the model, the target image is given an initial proportion, which is the degree of contribution of the target to the classification accuracy of the classifier. During the training process, each target image of the corresponding weight changes according to certain rules. Finally, each target image in the trained model also corresponds to a specific proportion.

The following is the basic process of the Adaboost algorithm:

1. Initialize all training samples with weights of $1/N$, where $N$ is the total number of samples.
2. Train weak classifiers and get the corresponding weight $\alpha_m$, set the number of weak classifiers to $m$.
   1. Train weak classifier $y_m(x)$ and make the weight error function $\varepsilon_m$ minimum. The $\varepsilon_m$ expression is as follows:

$$ \varepsilon_m = \sum_{n=1}^{N} \alpha_n^{(m)} I \left(y_m(X_n) \neq y_i\right) $$

2. Calculate the specific gravity $\alpha$, an expression for the weak classifiers is as follows:

$$ \alpha_m = \ln \left\{ \frac{1 - \varepsilon_m}{\varepsilon_m} \right\} $$

Figure 6. HOG characteristic extraction
3. Update the corresponding weight of the sample.

\[
\omega_{n+1,i} = \frac{\omega_i}{Z_n} \exp\left(-\alpha_m y_m(x_i)\right), \quad i = 1, 2, \ldots, N
\]

(6)

Among them, the role of \(Z_m\) is a normalization factor whose role is to ensure that all of the \(\omega\) is 1.

(3) The final output of the classifier is:

\[
Y_m(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right)
\]

(7)

The AdaBoost algorithm is trained using the positive and negative samples of the classified component pins. Finally, the accuracy is calculated from the test set.

4. Deep learning method

The problem with the above machine learning is whether the input feature provides a good distinction between positive and negative samples. For many machine learning problems, feature extraction is a difficult task. On some complex issues, an effective set of features is designed by artificial means but this needs a lot of time and effort.

Since the artificial approach cannot extract the characteristics of the entity, we can use the more efficient method of in-depth learning. One of the core issues of in-depth learning is the automatic combination of simple features into more complex features. In-depth learning can be a layer of simple features gradually transformed into more complex features [7-11].

This paper classifies the images using the convolution neural network (CNN), which consists of two layers. One layer is the feature extraction layer where the input of each neuron is connected with the local acceptance domain of the previous layer, and the local feature is extracted. Once the local feature is extracted, its positional relationship with other features is also determined. The second layer is the feature mapping layer where each calculation layer of the network is composed of multiple feature maps and each feature map is a plane; the weights of all neurons on the plane are equal. The feature mapping structure uses the sigmoid function, which affects the function nucleus, as the activation function of the convolution network so that the feature map has the displacement invariance. In addition, since the neurons share weights on a map surface, the number of free parameters of the network is reduced. The classical convolution neural network model includes the Le-Net5 model [12] and the Inception-v3 model [13].

The convolution neural network used in this paper is similar to the Le-Net5 model.

First layer: convolution layer

The input is the original image pixel, the input layer size is \(24 \times 24 \times 1\). The first convolution layer filter has a size of \(5 \times 5\), a depth of 6, and no full 0 padding. The output matrix is \(20 \times 20 \times 6\).

Second layer: pool layer

The input of this layer is the output of the first layer, which is the \(20 \times 20 \times 6\) node matrix. This layer uses a filter size of \(2 \times 2\) and the output matrix size of the layer is \(10 \times 10 \times 6\).

Third layer: convolution layer

The input matrix for this layer is \(10 \times 10 \times 6\), using a filter size of \(3 \times 3\), a depth of 16, and no full 0 padding. The output matrix is \(8 \times 8 \times 16\).

Fourth layer: pool layer

The input matrix of this layer is \(8 \times 8 \times 16\), the filter size is \(2 \times 2\), and the output matrix size of this layer is \(4 \times 4 \times 16\).

Fifth layer: whole connection layer

The upper output \(4 \times 4 \times 16\) matrix straightened into a vector. The output node of this layer is 80.

Sixth layer: whole connection layer

The input node of this layer is 80, and the output is 32 nodes.
Seventh layer: whole connection layer
The input node of this layer is 32, and the output is 2 nodes.

The above is the establishment of the CNN model used in this paper. The loss function is expressed by the cross entropy, which represents the proximity of the output vector and the expected vector. Given two probabilities \( p \) and \( q \), the expression is:

\[
H(p, q) = - \sum_x p(x) \log q(x)
\]  

The parameters in the neural network are adjusted by the back propagation algorithm [14] and the gradient descent algorithm. Finally, test the trained network with the test set to calculate the accuracy rate.

5. Analysis of results

5.1 Analysis of machine learning methods

The test accuracy rate of a small training sample set of 50 with a negative sample of 50 for training classification produced the following results. As table 1 shows, this is the test set accuracy.

| Algorithm      | Accuracy rate% |
|----------------|----------------|
| HOG+SVM        | 95.8%          |
| HOG+AdaBoost   | 96.2%          |

As the training effect is not ideal, the sample is increased to 500 with the number of negative samples also 500. As table 2 shows, this is the test set accuracy.

| Algorithm      | Accuracy rate% |
|----------------|----------------|
| HOG+SVM        | 96.1%          |
| HOG+AdaBoost   | 96.3%          |

Although the effect of increasing the sample set is effective, it is not significant. Using all the training set of 5800 samples with 6200 negative samples, the test set accuracy is as the table 3.

| Algorithm      | Accuracy rate% |
|----------------|----------------|
| HOG+SVM        | 96.2%          |
| HOG+AdaBoost   | 96.5%          |

It can be seen that the difference in effect of using different classification methods is not disparate. The reason for this is probably due to inaccuracies in the sample itself, as well as to the HOG operator not having extracted the classification characteristics well.

5.2 Analysis of in-depth learning methods

Although the positive sample of 5800 and negative sample of 6200 is basically satisfied for the depth of the neural network. The accuracy of the test set is as the table 4.

| Algorithm | Accuracy rate% |
|-----------|----------------|
| CNN       | 98.9%          |

The results of machine learning indicate that the HOG operator is not the best feature of classification.

6. Summary

Rather than a proposal to replace traditional methods of image processing with machine learning and in-depth learning, this paper simply presents the possibility of applying both as complementary methods. For example, since the greatest difficulty with the traditional machine learning is finding the best classification of the eigenvector, in-depth learning can be used to extract relatively good
characteristics and thus achieve a better classification result.

**Acknowledgement**
The authors gratefully acknowledge the support provided by the National Nature Science Foundation of China under NSFC Grant 51375118, as well as the support from Shenzhen Government Fund JCYJ20160330161701343 and JCY20170413105740689.

**References**

[1] Zhang W Y, Zhao Q S, Wang D Y. Actualities and Developing Trend Of Machine Vision[J]. Journal of Zhongyuan University of Technology, 2008.

[2] Kassim A A, Zhou H, Ranganath S. Automatic IC orientation checks[J]. Machine Vision & Applications, 2000, 12(3): 107-112.

[3] Kim H T, Song C S, Yang H J. Algorithm for automatic alignment in 2D space by object transformation[J]. Microelectronics Reliability, 2006, 46(1): 100-108.

[4] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]/ IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2005: 886-893.

[5] Cortes C, Vapnik V. Support-Vector Networks.[J]. Machine Learning, 1995, 20(3): 273-297.

[6] Freund Y, Schapire R E. Experiments with a new boosting algorithm[C]/ Thirteenth International Conference on International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. 1996: 148-156.

[7] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation, 2014, 18(7): 1527-1554.

[8] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436.

[9] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82-97.

[10] Schmidhuber J. Deep learning in neural networks: an overview[J]. Neural Networks, 2014, 61: 85-117.

[11] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep learning[M]. MIT Press, 2016.

[12] Lécun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[13] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[J]. 2015: 2818-2826.

[14] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533-536.