

Faster R-CNN Based Defect Detection of Micro-precision Glass Insulated Terminals

Qunpo Liu, Mengke Wang, Zonghui Liu, Bo Su

*School of Electrical Engineering and Automation, Henan Polytechnic University, 2001 Century Avenue
Jiaozuo, (454003), Henan, P.R.China,*

Hanajima Naohiko

*College of Information and Systems, Muroran Institute of Technology, 27-1 Mizumoto-cho, Hokkaido
Muroran-shi, Hokkaido (050-8585), Japan*

*E-mail: lqpnny@hpu.edu.cn, , hana@mondo.mech.muroran-it.ac.jp
www.hpu.edu.cn, www.muroran-it.ac.jp*

(leave space here should be 4 lines between www.university_name.edu and Abstract)

Abstract

Micro-precision glass insulated terminals (referred to as glass terminals) are the core components used in precision electronic equipment. As glass terminal, its quality has a huge impact on the performance of precision electronic equipment. Due to limitations in materials and production processes, some of the glass terminals produced have defects such as missing blocks, bubbles, and cracks. At present, it is difficult to ensure product quality and production efficiency with manual inspection methods. However, the defect characteristics of glass terminals are quite different, and it is difficult for traditional defect detection technology to design an ideal feature extractor for detection. Therefore, this paper proposes to use deep learning technology to detect missing blocks. First, preprocess the sample pictures of missing block defects of glass terminals, and then train the deep learning network based on Faster RCNN. According to the test results, the algorithm has an accuracy of over 90% in detecting missing defects in glass terminals.

Keywords: Micro-precision Glass Insulated Terminal; Faster R-CNN; Missing Block Detection

1. Introduction

Micro-precision glass insulated terminals are the core components used in precision electronic equipment. The glass terminal has a small size and requires high accuracy. Defects in the glass terminals would bring serious losses and consequences¹. The difficulties in defect detection are mainly three points²: (1) The complex imaging background of the defect contains a variety of interference noise; (2) The shape, size, and location of defects are diverse; (3) The difference in the characteristics of defects of the same type is large, and the difference in the characteristics of defects of different types is small. Therefore, this paper proposes to use deep learning technology to detect missing blocks³.

2. Image acquisition and annotation

The equipment used for image acquisition and photography is the CCD 540TVL (High Resolution B / W Black and White Camera) carried by the MVP400CNC automatic image measuring instrument; the working distance is 86mm; The glass terminal image shooting method adopts the vertical shooting mode of the eight-zone light source; the collection condition is strong light; the magnification during shooting is 1.0*30.77. This article uses labeling software for labeling. After labeling, an xml file will be generated, which can be used for Faster RCNN⁴ training.

3. Faster RCNN algorithm

3.1. Feature extraction network

The feature extraction network used in this article is ResNet50, and its structure is mainly composed of Conv Block and Identity Block^{5,6}. Among them, the Conv Block cannot be directly connected to the network. Its main function is to change the dimension of the network, and the dimensions of the input and output results are different. The identity block can be connected to the network, and its main function is to deepen the network depth, and the input and output results have the same dimensions.

ResNet50	
7×7 64,Conv layer, stride 2	
3×3 Max pool, stride 2	
$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 64 \\ 1 \times 1 & 256 \end{bmatrix}$	×3
$\begin{bmatrix} 1 \times 1 & 128 \\ 3 \times 3 & 128 \\ 1 \times 1 & 512 \end{bmatrix}$	×4
$\begin{bmatrix} 1 \times 1 & 256 \\ 3 \times 3 & 256 \\ 1 \times 1 & 1024 \end{bmatrix}$	×6
$\begin{bmatrix} 1 \times 1 & 512 \\ 3 \times 3 & 512 \\ 1 \times 1 & 2048 \end{bmatrix}$	×3
Average pool	

Fig.1 ResNet50 network structure

ResNet50 is mainly composed of residual blocks. Its structure is as shown in Figure 1. First, it performs a convolution operation with 64 convolution kernels of size 7*7 and step size of 2, and then performs a maximum pooling with a pooling kernel size of 3*3 Then, after going through 3, 4, 6, and 3 residual blocks, respectively, 1*1 and 3*3 represent the size of the convolution kernel of the residual block, and 64, 256 and so on represent the number of convolution kernels, respectively.

3.2. RPN

RPN is the region generation network. By using d n*n window sliding on the shared feature map extracted by the feature extraction network⁷, it is mapped into a d-dimensional feature vector, and k Anchors are generated on each pixel. These anchors perform the regression (reg)

and classification (cls) operations of the prior box respectively. Assuming that there are k anchors for each pixel on the shared feature map, it is necessary to predict these k pixels, and the classification operation can get 2k outputs, Indicates whether the k area contains the target. The regression operation refers to the calculation of continuously approaching the label frame with k anchors. This operation can get 4k outputs. The k anchors get the center point coordinates and width and height information of the suggested frame (x,y,w,h), the process of getting the suggestion box from anchor is also called anchor parameterization. Taking a pixel of the current shared feature map as the center, k anchors with different aspect ratios and sizes are generated, and their aspect ratios and sizes are represented by ratio and scale, respectively. For a feature vector of size w*h, there are a total of w*h*k anchors. The structure of the RPN network is shown in Figure 2.

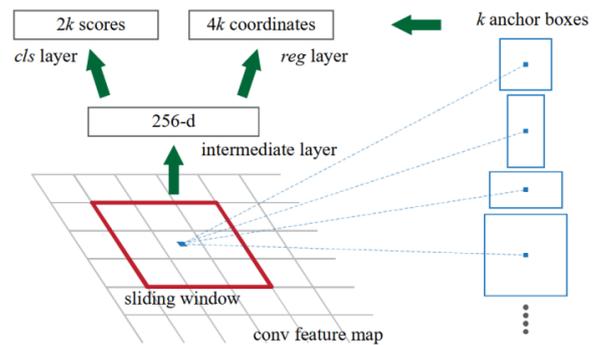


Fig.2 RPN network structure

3.3. RPN loss function

The overall loss function of RPN is

$$L(\{p_i\}, \{p_i^*\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where, i represents the index of the anchor, p_i and p_i^* respectively indicate that the i -th anchor contains the target predicted value and the anchor tag value. When the i -th anchor contains the target, $p_i^*=1$, and $p_i^*=0$ in other cases. t_i and t_i^* respectively represent the position and scale information of the suggestion box and the label box,

L_{cls} represents the classification loss, that is, the log loss of the anchor corresponding to the target category, and L_{reg} is the regression loss, that is, the loss of the suggestion box obtained from the candidate box, definition for

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (2)$$

where, R is the robust loss function (S_{L1}), that is, using Smooth_L1 to calculate the loss value.

It can be seen from $p_i^* L_{reg}$ that the regression loss can be calculated when $p_i^*=1$, and there is no regression loss in other cases. Among them, N_{cls} represents the selection of N_{cls} anchors for RPN training, N_{reg} represents the shape of the shared feature map, and λ is equivalent to a scale factor, so that the weights of the classification loss and the regression loss are basically the same (generally $\lambda=N_{reg}/N_{cls}\approx 10$).

The regression process uses 4 coordinates:

$$t_x = (x - x_a)/w_a, t_y = (y - y_a)/h_a \quad (3)$$

$$t_w = \log(w/w_a), t_h = \log(h/h_a) \quad (4)$$

$$t_x^* = (x^* - x_a)/w_a, t_y^* = (y^* - y_a)/h_a \quad (5)$$

$$t_w^* = \log(w^*/w_a), t_h^* = \log(h^*/h_a) \quad (6)$$

where, x, y, w , and h respectively refer to the center coordinates, width, and height of the a priori box. The variables x, x_a , and x^* respectively refer to the x coordinates of the suggestion box, anchor a priori box, and label box (the same applies to y, w , and h).

In the RPN network, the obtained prediction frames need to be screened. This process uses IOU and non-maximum suppression methods. The following is an analysis of these two methods.

3.4. ROI Pooling

Its main function is to merge the shared feature map and the ROI (region of interest) extracted through RPN, and then divide the prediction frame into a set number of grids, and pool each grid, which is equivalent to The prediction boxes of different sizes in the shared feature map are pooled into feature maps of the same size, which is convenient for inputting subsequent networks for classification and regression.

As shown in Figure 3, the size of the prediction box output by the RPN is fixed to pool_w and pool_h (where pool_w and pool_h are both 7), which is equivalent to

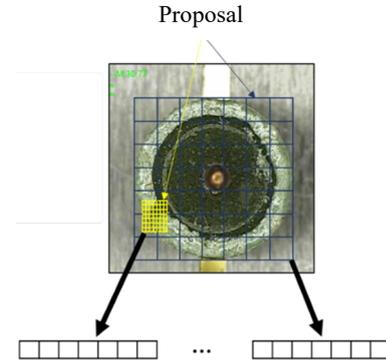


Fig.3 ROI Pooling working diagram

dividing it into 7*7 grids, that is, to achieve a fixed length output .

3.5. Classification network

The fully connected layer and the activation function are mainly used to judge and return the object category in the prediction frame to obtain a more accurate prediction frame, and its structure is shown in Figure 4.

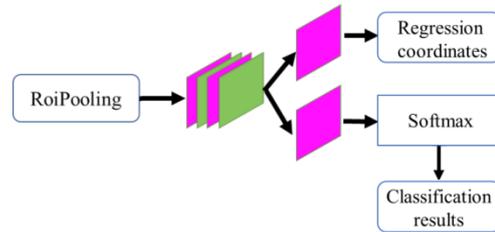


Fig.4 Classification network structure diagram

3.6. Loss function of Faster RCNN

Faster RCNN combines the loss functions of the classification model and the regression model, and a calculation formula can be used to calculate the overall loss of the network. The loss function expression after the network is merged is shown in equation (7).

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda(u \geq 1)L_{loc}(t^u, v) \quad (7)$$

where, p represents the probability that the corresponding anchor output is a certain category, u is the actual category label corresponding to the anchor, t^u represents the regression of the candidate frame to the scale factor corresponding to the predicted frame, and v is the

transformation parameter vector of the candidate frame regression.

$$L_{cls}(p, u) = -\log p_u \quad (8)$$

where, $L_{cls}(p, u)$ represents the classification loss, which is defined as the Equation (8), and the log base is e.

$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} S_{L1}(t_i^u - v_i) \quad (9)$$

where, $L_{loc}(t^u, v)$ represents the loss function of the prediction box obtained by regression of the candidate box, and S_{L1} represents the Smooth L1 loss function.

4. Experimental results

Using Faster RCNN to detect missing blocks and defects requires only 2 hours of training, and the number of training rounds is set to 50. It can detect whether more than 50 samples contain defects in 1 minute, with an accuracy rate of 98.03%, and with training as the number of rounds increases, there is still room for improvement in accuracy. Some results of this paper are shown in Figure 5.

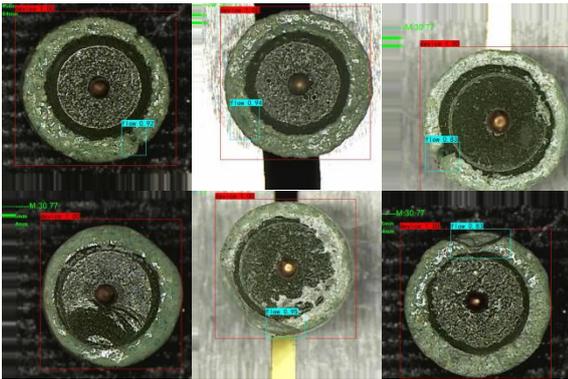


Fig.5 Example of Faster RCNN's Missing Block Defect Detection Results

5. Conclusion

The design of this article is supported by the working mechanism of CNN, and is designed for the implementation of the subsequent Faster RCNN and the detection of missing block defects. The detection process does not require manual feature extraction, but learns features based on the sample of the glass terminal to identify missing block defects and Perform calibration. With the expansion of the data set and the improvement of the model, there is still much room for improvement in the accuracy of missing block defect detection.

Acknowledgements

This work is partially supported by National Key Research and Development Project (2016YFC0600906), Innovation Scientists and Technicians Troop Construction Projects of Henan Province (CXTD2016054) and also supported by Innovative Scientists and Technicians Team of Henan Provincial High Education (20IRTSTHN019).

References

1. Q.Liu, M.Wang, G.Wang, R.Gao, N.Hanajima, Detection Algorithm of Porosity Defect on Surface of Micro-precision Glass, Encapsulated Electrical Connectors, Journal of Robotics, Networking and Artificial Life, 3 (2020) 212-216
2. Q.Liu, M.Wang, and N.Hanajima, Defect Sample Generation System Based On DCGAN for Glass Package Electrical Connectors, Proceedings of 2019 Chinese Intelligent Systems Conference, (2020)
3. WANG Z, ZHU D. An accurate detection method for surface defects of complex components based on support vector machine and spreading algorithm[J]. Measurement, 2019, 147: 106886.
4. Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J], Neural Information Processing Systems(NIPS). 2015.
5. Wang T, Chen Y, Qiao M, et al. A fast and robust convolutional neural network-based defect detection model in product quality control[J]. The International Journal of Advanced Manufacturing Technology, 2017.
6. He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C], IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2016.
7. Ren S, He K, Girshick R, et al. Object Detection Networks on Convolutional Feature Maps[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(7): 1476-1481.
8. Y.Jia, Alternative proofs for improved LMI representations for the analysis and the design of continuous-time systems with polytopic type uncertainty: a predictive approach, IEEE Transactions on Automatic Control, 48(2003) 1413-1416
9. Y.Jia, General solution to diagonal model matching control of multi-output-delay systems and its applications in adaptive scheme, Progress in Natural Science, 19(2009), 79-90.