CNN-Based Surface Defect Detection of Smartphone Protective Screen

Hailang Chen
School of EIE
Heyuan Polytechnic
Heyuan, Guangdong 517000, China
Email: chenhailang522@foxmail.com

Abstract — With the advancement of the intelligent era of Industry 4.0, in order to solve the problem of low-efficiency and high miss-detection rate using manual visual inspection for the detection of surface defects on smartphone protective screens, surface defects on smartphone protective screens based on convolutional neural networks are introduced as the detection method. This detection method exhibits excellent feature extraction capabilities and powerful target classification performance during the preprocessing, model design, model training, and detection of smartphone protective screens images. Experimental results show that the detection method can achieve accurate detection of defects such as punctures, bright spots, and scratches on the surface of the smartphone protective screens. The detection verification rate is as high as 98%, and the accuracy is high, which meets the actual needs of enterprises.

1. Introduction

With the continuous improvement of people's living standards, smartphones have gradually become a necessity for everyone's daily entertainment, life and work. The quality of the protection screen is directly related to the overall quality of the phone. Therefore, the detection of the appearance of the mobile phone protective screen is one of the important links in the intelligent production process. However, the current major smartphone manufacturers mostly use the naked eye to detect the defects of mobile phone screens [1]. The cost of manual testing is high, inefficiency and subject to human subjective factors; the accuracy of testing is difficult to prove. The detection of surface defects of smartphone protection screens based on the machine perspective will be a trend for future development.

The research on surface defect detection is an important research direction in machine vision. Experts and scholars have conducted research on surface defect detection in different situations, and have proposed many defect detection methods. Vasilic utilizes Prewitt algorithm and threshold selection algorithm based on histogram to realize the detection of scratches on the ceramic surface [2]. Fan Wei adopted a processing method based on the frequency-domain graph smooth band-pass filtering to realize the detection of scratches on the surface of automobile body parts [3]. Xuechen Sun et al. used domain weighted segmentation to extract surface defects to achieve the detection of camshaft surface defects [4]. Zhiyong He et al. proposed an algorithm for searching for surface defect areas using the gradient image variance distribution, and applied
the Otsu method to segment images in local areas with surface defects [5]. This method can quickly and automatically detect small surface defects from surface images with little background changes.

During the production process of the smartphone protection screen, defects such as bright spots and brightness may occur. There are punctures, scratches, etc. during transmission. The lack of detection on the surface of smartphone protective screens based on convolutional neural networks (CNN) aims to take advantage of CNN's advantages in computer vision technology detection, reduce human resources, reduce production costs, and mass-produce smartphone protective screens. The big data processing and analysis of the process provide new ideas and promote the agglomeration and development of the smartphone industry.

2. The design of defect detection method

2.1 Process design

The surface defect detection of the smartphone protective screen based on the convolutional neural network is mainly divided into three processes: image preprocessing, model design and training, and detection, as shown in Figure 1. The data sample comes from the image taken by the high-precision line camera on the production line. The defects include scratches, scuffings, punctures, bright spots and light leakage. The process of preprocessing is to scan, segment and save the image.

![Figure 1: Design of detection process for surface defects of mobile phone protective screen](image)

2.2 The feature classification of data set defects

Defects of smartphone protection screens are named differently according to different causes, but many different types of defects have similar defect patterns. As shown in Figure 2, there are three pictures of scratches or scuffings. These two types of defects are similar in defect morphology and are linear continuous defects. As shown in Figure 3, bright spots, punctures, and brightness defects do not differ greatly in morphology, so they are collectively referred to as point defects.
Scratches Scuffings Punctures

Figure 2 Linear continuous defects

Bright-spot Stub Brightness

Figure 3 Point defects

After the initial segmentation, 960 defect samples were obtained. In view of the needs of convolutional neural network training for big data, this paper expands to 9615 pieces of data by means of automatic random cropping, rotation, mirror transformation and changing the image contrast, and randomly distributes the training set, verification set and test set according to the proportion. Finally, the Imdb training file of each data set and the average file of each data set are automatically generated on the Caffe platform. The test set distribution diagram is shown in Figure 4.

3. Model design and training

3.1 Model design
Currently, the mainstream CNN models, such as VGGNet, GoogLeNet, and ResNet are designed for the ImageNet data set, which requires a huge network model and a large number of parameter designs for learning complex features in natural images. Due to the advancement of modern technology, the surface defects of the smartphone protection screen have almost one kind of defect on one sample, and the defect characteristics are relatively simple. Therefore, for the unity of the training samples of the smartphone
protection screen, in this experiment, 22 layers of GoogLeNet with moderate parameters are selected for fine-tuning. After removing the inceptionB layer and the inceptionB layer, the fully connected layer and Softmax used for weighted classification Floor. Because the parameters of the fully connected layer are more redundant, there are too many parameters (about 80% of the entire network parameters), which reduces the training speed and is easy to overfit. The commonly used solution is to use the dropout random drop mechanism, but dropout cannot solve the problem in essence, but the random nature of dropout may drop some important neurons [6]. Therefore, this experiment uses global average pooling instead of the fully connected layer to regularize the structure of the entire network, giving each neuron the actual category meaning to prevent overfitting (see Figure 5). Finally, the 1000-dimensional column vector of the softmax layer is adjusted to a 3-dimensional vector for three classifications, which makes the network more lightweight and accelerates the training speed of the model.

3.2 Model training

This experiment is completed on the Ubuntu16 system to build the Caffe framework. In order to prevent the situation that the training is difficult to converge due to the large difference in the distribution of training data, the experiment uses a data normalization method, and the normalization function is as formula (1).

$$\bar{x} = \frac{x - \text{Min}}{\text{Max} - \text{Min}}$$

(1)

Where $\bar{x}$ is the standardized input data, Min is the minimum value of the training data set, and Max is the maximum value of the input data.

The training uses a stochastic gradient descent algorithm to update the network parameters. The learning rate is a coefficient that guides the SGD gradient to update the network weights. The lower the learning rate we set, the slower the rate of change of the loss function will be. The size of the learning rate affects the convergence rate of the network. Although a too small learning rate can ensure that the network optimization does not miss any local minimum, it will also increase the training time. Therefore, this paper uses a decay type learning rate to optimize the network training speed. The learning rate decay mechanism is shown in formula (2):

$$\eta = \text{base\_lr}.\gamma^{\frac{\lfloor \text{floor\_iter} \rfloor}{\text{stepsize}}}$$

(2)

Where $\eta$ is the learning rate, and $\text{base\_lr}$ is the initial learning rate of the network, which is a floating-point real number. The $\gamma$ parameter indicates the degree of change of the learning rate every time, and the value is generally a constant. $\text{Floor\_iter}$ is the current number of iterations, and $\text{stepsize}$ indicates the number of iterations where the learning rate changes once. In this paper, the $\text{base\_lr}$ set by experiment is 0.001, $\gamma$ is 0.1, $\text{stepsize}$ is 337, and the learning speed transformation is shown in Figure 6. The momentum is set to 0.9.
The training results are shown in Figure 7. The orange curve (val) represents the accuracy of each round of the model (epoch), the green is the loss value of each round of the verification set, and the blue (loss) represents the loss curve of the model. From the trend of the curve, it can be judged that the verification accuracy of the model is gradually increasing, which tends to be stable when the epoch is 60, but still maintains the upward trend. The loss of the model is declining. It can be seen from the figure that the highest accuracy of the verification set is 98%.

4. Experimental results and analysis
In order to verify the training results of the model, this paper made five experiments on networks with different training sets, different learning rates, and different training batches under 963 test sets. The distribution of the test set is shown in Figure 4. All positive samples of training set, point defect samples and line defect samples are roughly distributed in a ratio of 1:1:1. Among them, the detection rate is the accuracy rate of the actual test in the test set, and its calculation formula is as formula (3), \( N_0 \) is the number of positive samples detected in the test set, \( N_1 \) is the number of point defect samples detected in the test set, and \( N_2 \) is the number of line defect samples detected in the test set, \( N_t \) is the number of test sets. The false detection rate is the probability that the test result of the test set is completely wrong, and the calculation formula is as formula (4).
\[
A = \frac{N_0 + N_1 + N_{02}}{N_f}
\]

\[
A = \frac{N_{0+1,2} + N_{1,2-0}}{N_p}
\]  

\(N_{0\rightarrow1,2}\) is the number of positive samples detected as negative samples, and \(N_{1,2\rightarrow0}\) is the number of negative samples detected as positive samples. In particular, the samples that detect errors between point and line defect samples are not included in the calculation of the false detection rate. The experimental results of the five experiments are shown in Table 1.

Table 1 The experimental results of the five experiments

| Experiment | Data set | Learning rate | epoch | Detection rate | False detection rate |
|------------|----------|---------------|-------|----------------|----------------------|
| Experiment 1 | 960      | 0.005         | 600   | 83.3%          | 6.0%                 |
| Experiment 2 | 2636     | 0.01          | 600   | 90.1%          | 4.0%                 |
| Experiment 3 | 9640     | 0.01          | 1800  | 99.5%          | 0.5%                 |

Experiment 1: In the case of 960 training samples, the learning rate is 0.005, the bottom batch size is 30, and the epoch is 600. The verification rate of the model is 89.6%, the actual test detection accuracy rate is 88.3%, and the false detection rate 6.0%. After the analysis, it was decided to expand the data set and increase the number of training samples:

Experiment 2: When the training samples are increased to 2636, and the parameters obtained are the same as in Experiment 1, the model's verification rate is increased to 95.5%, and the detection rate is increased to 89.2%. On the basis of Experiment 1, the learning rate was increased to 0.01

Experiment 3: The actual detection effect was improved when the learning round was increased to 1800 times. The model verification rate was 98.0%, the detection rate was 99.5%, and the false detection rate was 0.5%. The detection results are shown in Table 2. 0 is a positive sample, 1 is a point defect, and 2 is a line defect.

Table 2 The experimental results of experiment 3

|      | 0   | 1   | 2   | Pre-class accuracy |
|------|-----|-----|-----|-------------------|
| 0    | 370 | 2   | 2   | 98.65%            |
| 1    | 0   | 290 | 0   | 100%              |
| 2    | 0   | 0   | 298 | 100%              |

After doing many experiments as above, all have shown that increasing the epoch on the basis of the existing data set does not greatly improve the actual detection rate, but will increase the machine learning time.

5. Conclusion

Aiming at the problem that the automatic detection of the appearance of the smartphone protective screen is more difficult in the production process, this experiment designs a CNN-based classification detection method for the appearance defect of the smartphone protective screen. This method divides the smartphone protection screen image into 256x256 pixel image samples to adapt to network training and detection. Then, the training data was expanded using data amplification related technologies. Secondly, by adjusting the parameters multiple times, deleting the layers with more parameters, and selecting the average pooling layer instead of the fully connected layer to classify the defects. Finally, through the fine-tuning of the parameters of the five training experiments, the network can ensure the accuracy of the surface defect detection of the
smartphone protective screen when the overall parameters are reduced by half. Experimental data shows that the method can effectively detect defects such as stabs, bright spots, brightness, scratches, and scuffings on the appearance of the mobile phone protective screen. Taking into account the differences in detection targets and product defects, by adjusting the experimental parameters, the accuracy of the CNN-based smartphone protective screen surface detection will also have a certain impact. Therefore, the accuracy of the learning model of the sample has not been compared, and I look forward to further discussion in the future.

Acknowledgments

This work was supported in part by the Fund for “Promoting the reform of higher education by using big data technology, Energizing teachers and students to explore the future” (FPE) and Foundation of Guangdong Education Department. We would like to thank all sponsors and reviewers for their detailed comments on our paper.

References

[1] Tang B, Kong J Y, Wu S Q. “Feature Extraction and Recognition Based on Machine Vision Application in Lotus Picking Robot,” Journal of Image and Graphics, vol.22, no.12, pp.1640-1663, 2017.
[2] Vasilic S, Hocenski Z. “The Edge Detecting Methods in Ceramic Tiles Defects Detection,” in Security and Privacy, 2006. SP 2006. IEEE Symposium on. IEEE, 2006, pp. 469–472.
[3] Qin F W, Gao N N. “Fine-grained leukocyte classification with deep residual learning for microscopic images,” Computer Methods and Programs in Biomedicine, vol.5, no.24, pp.243-252, 2018.
[4] Li X C, Jiang X N. “Camshaft surface defect detection system based on machine vision,” Infrared and laser engineering, vol.42, no.6, pp.1647-1653, 2013.
[5] He Z Y, Sun L, Rui Y N. “Machine vision detection method for tiny surface defects,” Journal of Applied Sciences, vol.30, no.5, pp.531-537, 2012.
[6] rivastava N, Hinton G, Krizhevsky A. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” Journal of Machine Learning Research, vol.15, no.1, pp.1929-1958, 2014.