

ScratchNet: Detecting the Scratches on Cellphone Screen

Zhao Luo^{1,2}, Xiaobing Xiao³, Shiming Ge^{1,2(✉)}, Qiting Ye^{1,2},
Shengwei Zhao^{1,2}, and Xin Jin⁴

¹ Institute of Information Engineering, Chinese Academy of Sciences,
Beijing, China

² School of Cyber Security, University of Chinese Academy of Sciences,
Beijing, China

`geshiming@iie.ac.cn`

³ School of Software and Microelectronics, Peking University, Beijing, China

⁴ Department of Computer Science and Technology,
Beijing Electronic Science and Technology Institute, Beijing, China

Abstract. In the process of cellphone screen manufacture, equipment failures and human errors may lead to screen scratches. Traditional manual ways check scratches by human eyes, which often costs large manpower and time, but with poor effectiveness. To address this issue, we proposes an automated scratches detection method by cascading two main modules. First, the scratch filtering module detects big scratches and localizes small scratches candidates with a serial of low-level stages. Then, the scratches classification module applies a lightweight CNN model, ScratchNet, to identify each small scratch candidate whether it is real scratch or not. To train the ScratchNet, we build a Scratches on Cellphone Screen (SCS) dataset with 50K samples in 5 categories. The experimental results on SCS testing set show that the proposed method achieves an accuracy of 96.35% on classifying small scratches, which outperforms the LeNet model and the other classifiers.

Keywords: Scratch detection · Product quality inspection
Cellphone screen · Convolutional neural networks

1 Introduction

Cellphones are increasingly common. Every mobile phone brand rolls out new products and it is more frequently that people replace their own mobile phones. At the same time, handset firms have continued to pursue innovation in the material and shape of phone screen. It sets a higher request to the technology of defecting scratches on the cellphone screen, so how we do that quickly and well? Conventional manual detecting methods can't meet the production requirements. Along with the development of industrialization and the arrival of the Industry 4.0 and AI 2.0, machine vision is more and more prosperous, which is

used in all walks of life. For example, rail track flaw detection [3], glass defects inspection [14, 15, 22], PCB coating detection [4, 12], LED inspection [16] and printing quality checking [17, 21]. It brings about huge improvement to production speed and product quality. It also reduces the cost of labor and enhances enterprise competitive power. From a macro view, it increases the using rate of social source and frees people from manual labor.

The main contributions of this paper are three folds:

- We propose a scratches inspection method to detect the scratches on the cellphone screen by cascading low-level image processing and a lightweight convolutional neural network (CNN) classifier, which can provide fast and accurate scratch detection performance.
- We build a new screen scratch dataset, SCS, to facilitate the scratch detection model.
- We design a lightweight CNN model, ScratchNet, which achieves an accuracy of 96.35% on classifying small scratches, which outperforms the other CNN models and other classifiers.

2 Related Work

In this part, we introduce some related works, mainly including surface defects inspection and convolutional neural networks.

2.1 Surface Scratches Inspection

In [15], Peng uses downward threshold to remove the background pattern like stripes firstly. Then he segment deflections by fixed threshold method and OTSU. The method work well in common defects, but it gets poor results when inspecting small and slight optical defects. In detecting PU-packings, Choiu [1] combines radius inspection method and projection, which means it can detect seven major types of flaw. However, the speed of inspection isn't enough stability and accuracy rate only reach 90.1%. Zhao and Kong [22] take Canny, Binary Feature Histogram (BFH) and AdaBoost when analyzing the glass quality. However, this method is limited to inspect two types of defects and it's measuring precision is about 25–100 pixels. With the development of machine vision, the way of detecting defects emerge one after another. Liang [11] applies PCA and Redundant Dictionary to inspect flaws on touch screens. Choi [2] adopt the Phase Only Transform (PHOT) in detecting of surface defects. Soukup [19] takes CNN to inspect metal surface defects. What's more, some people capture surface defects by other ways, for example, McGrail [13] uses standard medical ultrasound scanners to collect information of insulating material flaws.

2.2 Convolutional Neural Networks

LeCun and Bengio recognize handwritten digits [10] by LeNet model and got better performance. Goodfellow designs CNN to identify Google StreetView

house number [7]. Ciresan applies CNN to traffic sign recognition benchmark [5]. Besides, CNN combined with other methods make a unprecedented breakthrough in driverless cars and faces recognition. Not only the extension of its application, but also the innovation of itself, such as Krizhevsky' AlexNet [9], Simonyan' VGGNet [18], Szegedy' GoogleNet [20] and He' ResNet [8] are more and more powerful.

3 Dataset and Method

3.1 Dataset

Data collection is very meaningful and important in facilitating scratch detection. In this paper, we build a Scratch on Cellphone Screen (SCS) dataset. First, we take a lot of cellphone screen images in various lighting environments and capturing poses. Then, through human checking, we crop a large scale of small scratches as well as background patches from screen images. In addition, according to the scratch shape in blob or line as well as saliency degree in salient or weak, we group the examples into five categories: background examples, salient blob examples, weak blob examples, salient line examples, and weak line examples. Then, we clean data and ask 3 persons to perform cross-check in order to ensure the quality of the dataset. Finally, the examples are normalized into a unified size of 32×32 pixels, and divided into training set and testing set. In total, the dataset has 5K images including 4.6K images for collecting training

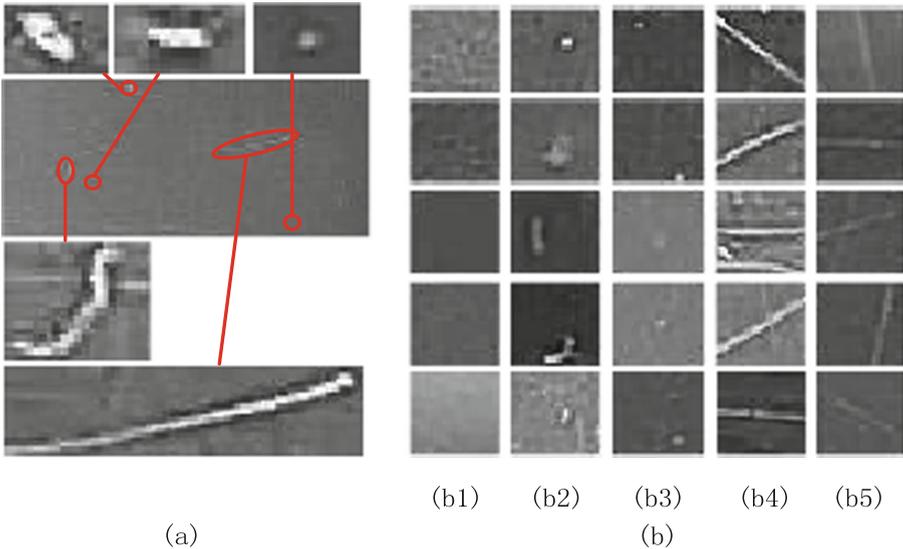


Fig. 1. Examples of scratches on the cellphone screen: (a) different categories of scratches are collected. (b) some examples of five scratch categories: (b1) background example, (b2) salient blob example, (b3) weak blob example, (b4) salient line example, (b5) weak line example.

set and the rest 4K images for testing. The training set includes five categories of examples: 12K background examples, 9K salient blob examples, 12K weak blob examples, 8K salient line examples, and 8K weak line examples. The testing images include five categories of scratches.

As shown in Fig. 1, scratches are diversity reflected on two main aspects: (1) the scratches have discrepant shapes, such as line-shape and blob-shape, and (2) their appearances are also very different due to diversified collection environments. Therefore, it will bring a great challenge to the scratch deflection algorithms.

3.2 Scratch Filtering

To speed up detection, we start with scratch filtering processing, which can first filter out the big scratches and make the processing focus on the more difficult small scratches. As shown in Fig. 2, we process the captured images via four main stages, including color transformation, binarization, morphologic processing and connected component analysis. After that, we can get some big scratches and small scratch candidates. In this manner, the big scratches are filtered out in the following processing, which greatly reduces the detection complexity.

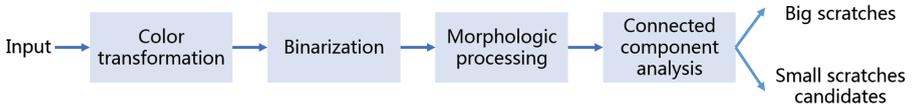


Fig. 2. Flowchart of scratch filtering. It could filter out the big scratches and make the processing focus on more difficult small scratch candidates.

Color transformation. Since detecting scratches directly on the colour image may have liter effects, we simplify the processing in the gray level image by investigating that the gray image has much litter liter effects. Therefore, the captured colour image could be transformed into gray image.

Binarization. In general, the cellphone screen is very smooth. If the scratches exist, it will generate gray level change in the regions of scratches. According to this fact, simply binarization by using a certain threshold will coarsely expose the scratches. Toward this end, we apply a threshold value to binarize the gray image, which segment the gray image into background and foreground (scratches). Suppose that $f(x, y)$ is gray value at point (x, y) and $g(x, y)$ is the value after segmentation, then

$$g(x, y) = \begin{cases} 255 & f(x, y) > T \\ 0 & f(x, y) \leq T \end{cases} \quad (1)$$

Morphologic processing. After binarization stage, the segmentation image often has discontinuous big scratches and very fractional small foregrounds. To address these issues, we adopt erosion and dilation operators in morphologic

processing, which could ensure the continuity of big scratches while efficiently eliminate very fragmentary scratch candidates.

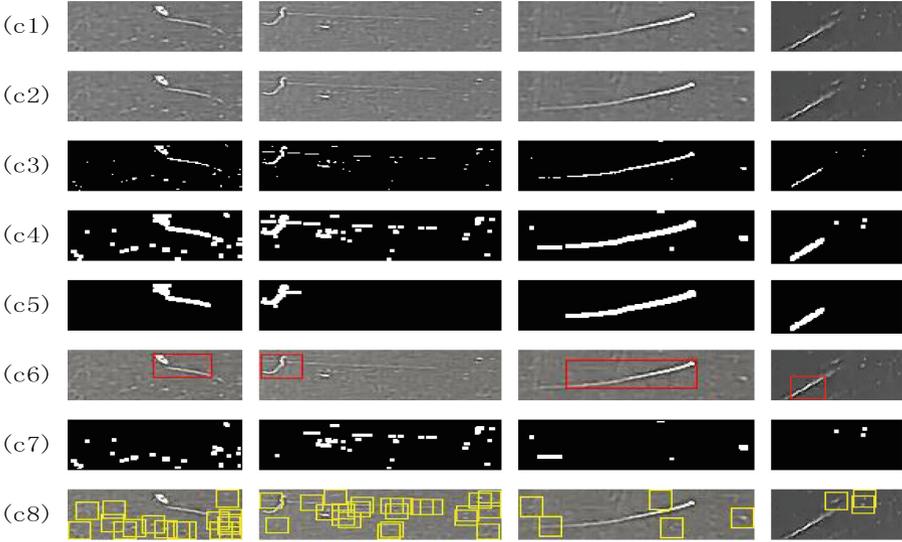


Fig. 3. Some examples of scratch filtering. (c1) original colour images, (c2) gray images, (c3) binary images, (c4) segmentation images, (c5) detected big scratches in segmentation images, (c6) detected big scratches in gray images, (c7) detected small scratch candidates in segmentation images, and (c8) detected small scratch candidates in gray images.

Connected component analysis. Then, we apply connected component analysis (CCA) to the filtered segmentation image. By using 8-connected region algorithm, we can label the connected regions and get their information, such as area and location. Suppose that there are n connected regions and their areas are represented as $\{S_i, i = 1, 2, \dots, n\}$. Accordingly, we could distinguish connected region by thresholding its area and classifying it as big scratch or small scratch candidate:

$$h(i) = \begin{cases} 1 & S_i > s \\ 0 & S_i \leq s. \end{cases} \quad (2)$$

where, s is area threshold value, h is a binary classification function. If $h(i) = 1$ the region S_i is considered as big scratch and otherwise small scratch candidate.

In actual production process, if some big scratches are detected on cellphone screens, the testing examples will be considered as unqualified. This process will accelerate inspecting process without further detection. The method of detecting small scratches sounds like Girshick's RCNN [6], but we can assume the rule that the candidate box of small scratch is fixed according to the raw image patches of

small scratch candidates. Figure 3 illuminates some examples of the scratch filtering. It shows the processing results in each stage. Note that scratch filtering shifts the task to classification of small scratch candidates, as shown in Fig. 3(c8).

3.3 Scratch Classification

After scratch filtering, the remaining small scratch candidates need to be identified. As shown in Fig. 3(c8), it is a challenging task due to the various appearance of small scratches candidates in weak lighting, shape deformation, and so on. To address these issues, we design a lightweight CNN, ScratchNet (see Fig. 4) to obtain robust representation of these small scratches, which can overcome these complex environment better than traditional methods. In details, our ScratchNet combines the structure of LeNet [10] and convolutional layers of VGGNet [18]. In this manner, our ScratchNet is not only as lightweight as LeNet [10], but also as strong as VGGNet [18]. As shown in Fig. 4, ScratchNet has eight layers, where the first six layers is convolution and each is a stack of two 3×3 conv to extract robust feature representation. After every convolutional layer, we apply a max pooling operation to lower dimensions of features. The last two layers respectively include one fully connection layers and one softmax layer to classify small scratches.

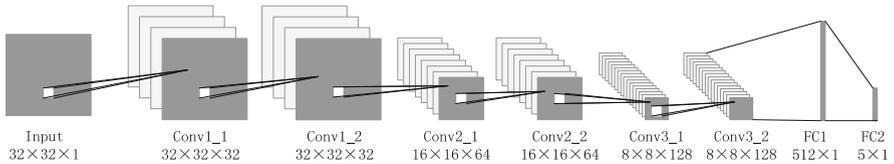


Fig. 4. The structure of ScratchNet. It includes three convolution layers, two fully connection layers and one softmax layer. Besides, the output of each layer is shown.

4 Experiments

We evaluate our methods by two experiments to demonstrate its effectiveness.

Firstly, we compare our ScratchNet with classic convolutional neural network, LeNet [10] on SCS. As shown in Fig. 5, not only ScratchNet is slightly better than LeNet in accuracy, but also ScratchNet has faster convergence speed than LeNet. The results on SCS by ScratchNet (See Table 1) demonstrate the effectiveness of our methods.

Secondly, we apply some machine learning methods on the SCS dataset. The results are given in Table 2. From the results, we can see that different methods have different performance. Obviously, other machine learning methods give lower classification performance than CNN. In addition, our ScratchNet has best results.

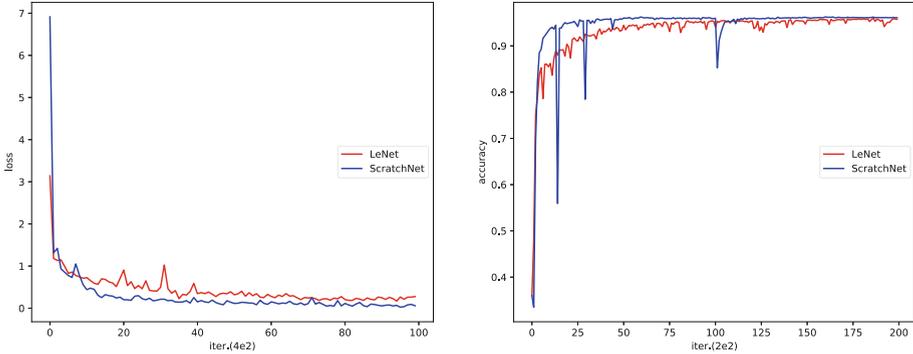


Fig. 5. The training loss (left) and accuracy (right) between ScratchNet and LeNet.

Table 1. Classification performance on SCS test set with ScratchNet.

Background	Notable batch	No notable batch	Notable line	No notable line
99.32%	0%	0.67%	0%	0%
0%	96.45%	2.26%	0.97%	0.32%
1.60%	1.04%	96.46%	0%	0.90%
0%	4.67%	0.33%	92.00%	3%
1.0%	0.67%	12.33%	1.67%	84.33%

Table 2. The results with five methods on the SCS datasets.

Random forest	Decision tree	Gradient boosting	LeNet	ScratchNet
83.89%	70.52%	73.15%	95.97%	96.35%

5 Conclusion

Our method is very practical in detecting scratches of cellphone screens. At first, if testing screens has big scratches, we can regard it as substandard screen and stop. Then we analyze candidates of small scratches as required. It improves the speed of detection, as well enhance the precision, cut down on hardware costs and advance quality. However, many existing CNNs are focus on relatively big objects and there is some room for improvement in small object inspection, so we can optimize our ScratchNet and apply ScratchNet in other issues. Except that, although we can find fixed candidates box quickly by some laws, our solution is not end-to-end just like Faster RCNN [6]. So, we can do further researches on the issue that how to introduce some useful rules into the CNN model.

Acknowledgments. This work is supported in part by the National Key Research and Development Plan (Grant No. 2016YFC0801005), the National Natural Science Foundation of China (Grant No. 61402463) and Open Foundation Project of Robot Technology Used for Special Environment Key Laboratory of Sichuan Province in China (Grant No. 16kftk01).

References

1. Chiou, Y.C., Li, W.C.: Flaw detection of cylindrical surfaces in PU-packing by using machine vision technique. *Measurement* **42**(7), 989–1000 (2009)
2. Choi, J., Kim, C.: Unsupervised detection of surface defects: a two-step approach. In: *IEEE International Conference on Image Processing*, pp. 1037–1040 (2012)
3. Clark, R.: Rail flaw detection: overview and needs for future developments. *NDT E Int.* **37**(2), 111–118 (2004)
4. Coulombe, A., Cantin, M., Bérard, L., Gauthier, J.: Method and system for detecting defects on a printed circuit board (2004)
5. Dan, C., Meier, U., Masci, J., Schmidhuber, J.: A committee of neural networks for traffic sign classification. **42**(4), 1918–1921 (1921)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation, pp. 580–587 (2014)
7. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. *Comput. Sci* (2013)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *International Conference on Neural Information Processing Systems*, pp. 1097–1105 (2012)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
11. Liang, L.Q., Li, D., Fu, X., Zhang, W.J.: Touch screen defect inspection based on sparse representation in low resolution images. *Multimedia Tools Appl.* **75**(5), 2655–2666 (2016)
12. Loh, H.H., Lu, M.S.: Printed circuit board inspection using image analysis. *IEEE Trans. Ind. Appl.* **35**(2), 426–432 (1999)
13. Mcgrail, A., Risino, A., Auckland, D.W., Varlow, B.R.: Use of a medical ultrasonic scanner for the inspection of high voltage insulation. *IEEE Electr. Insul. Mag.* **9**(6), 5–10 (2002)
14. Pei, K.: Study of on-line glass defect inspection system based on embedded image processing. *Electron. Meas. Technol* (2009)
15. Peng, X., Chen, Y., Yu, W., Zhou, Z., Sun, G.: An online defects inspection method for float glass fabrication based on machine vision. *Int. J. Adv. Manuf. Technol.* **39**(11–12), 1180–1189 (2008)
16. Perng, D.B., Liu, H.W., Chang, C.C.: Automated SMD LED inspection using machine vision. *Int. J. Adv. Manuf. Technol.* **57**(9–12), 1065–1077 (2011)
17. Shang, H.C., Chen, Y.P., Yu, W.Y., Zhou, Z.D.: Online auto-detection method and system of presswork quality. *Int. J. Adv. Manuf. Technol.* **33**(7–8), 756–765 (2006)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *Comput. Sci* (2014)
19. Soukup, D., Huber-Mörk, R.: Convolutional neural networks for steel surface defect detection from photometric stereo images. In: *Bebis, G., Boyle, R., Parvin, B., Koracin, D., McMahan, R., Jerald, J., Zhang, H., Drucker, S.M., Kambhamettu, C., El Choubassi, M., Deng, Z., Carlson, M. (eds.) ISVC 2014. LNCS, vol. 8887, pp. 668–677. Springer, Cham (2014)*. https://doi.org/10.1007/978-3-319-14249-4_64

20. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: *Computer Vision and Pattern Recognition*, pp. 1–9 (2015)
21. Verikas, A., Lundström, J., Bacauskiene, M., Gelzinis, A.: Advances in computational intelligence-based print quality assessment and control in offset colour printing. *Expert Sys. Appl.* **38**(10), 13441–13447 (2011)
22. Zhao, J., Kong, Q.J., Zhao, X., Liu, J., Liu, Y.: A method for detection and classification of glass defects in low resolution images. In: *International Conference on Image and Graphics*, pp. 642–647 (2011)