ISBN 978-81-936279-9-0
10th Int'l Conference on Innovations in Engineering, Technology, Computers & Applied Sciences (IETCAS-18)
Kuala Lumpur (Malaysia) July 23-25, 2018

Classify LCD Panel Defect Type with Panel Image and Structured Defect Feature Data

Mungso Kim[1], Minyoung Lee[1] , Minjeong An[1] and Hongchul Lee[1]

[1]Department of Industrial Engineering, Korea University

***Abstract*:** *Even though Convolutional Neural Network (CNN) makes outstanding performance in image classification recently, it is still hard work for CNN to classify defect class on panel image. Main reasons that make classification problem hard, are various defect size and relatively large meaningless pattern on panel background. To solve this problem, we propose deep and wide CNN architecture combining both panel image and defect feature data from background pattern elimination and computer vision techniques.*

Keywords: *Defect Classification, CNN, Pattern Elimination*

1. Introduction

Traditional automatic panel image classification used human selected feature with computer vision techniques. Zhang et al used morphological operations and thresholding [1], Kameyama et al [2] used image difference, and Lim et al [3] used defect shape description to get defect feature from image.

But recently, Convolutional Neural Network (CNN) is widely used for defect classification [4] and localization [5]. CNN is one type of Artificial Neural Network (ANN) which is specified for image classification first suggested by LeCun et al [6]. CNN get famous because it can automatically finds feature of labelled image and its performance is outstanding compared to other computer vision based feature.

Each methods has its strong points, so we tried to combine both human selected features and CNN features. Collaboration of two methods increased classification accuracy of four types of LCD panel defect which is GA, IP, OP, SA. The description of defects is as follows.

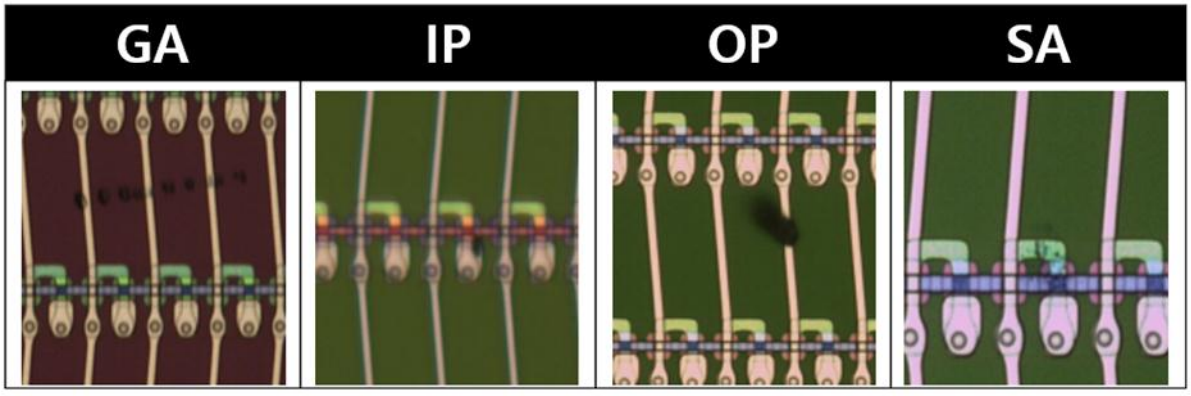

Fig. 1: Typical Defect Shape of Each Classes

GA : Defects are scattered in a line like scratch

IP : Defects are on the background pattern

OP : Part of defect is blur

SA : Defects are scattered like spray

We used shape detection techniques after background pattern elimination for structured data. Raw panel image has 2048x2048 resolution but we reshaped into 256x256 size for CNN input.

2. Methodology

2.1. Background Pattern Elimination

As seen at figure1, defect is tiny in original data and background pattern is a main factor that decrease classification accuracy. So we need to eliminate background patterns to get better classification result. But those are reasons that makes background elimination work harder.

The location of patterns is all different between images. Also even on single image, color saturation of background is slightly different between middle and edge. But with a method stated below, we successfully eliminate pattern and convert original defect image into binary image

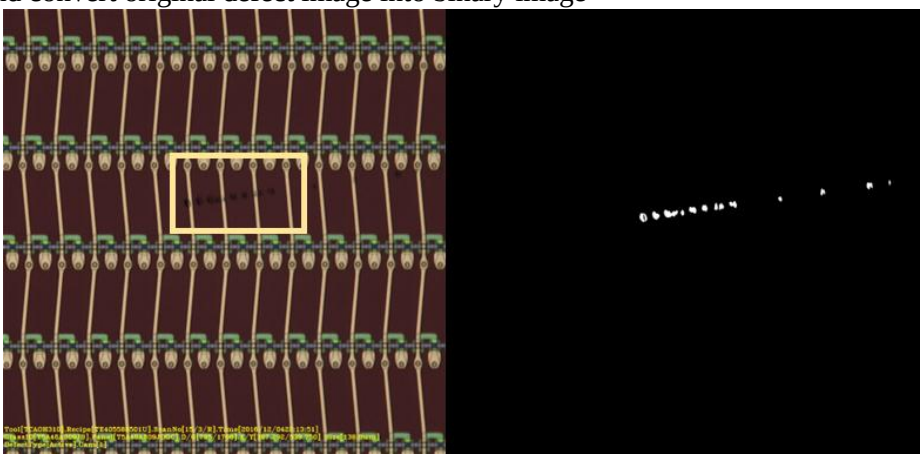

Fig. 2: Original Defect Image (Left) and Binary Image after Background Pattern Elimination (Right)

2.1.1. Find Pitch

Pitch refers to cycle pixel length of one unit of pattern. To find pitch, we used template matching on topleft partition image with 1/10 width, 1/4 height of original image as template (T) and topleft partition image with 1/4 width, 1/4 height as target image (I). Measure of template matching similarity 'R' is calculated as normalized dot product which is descripted as formula (1).

$$R(x) = \frac{\sum_{x'}(T(x',y') \cdot I(x'+x,y'))}{\sqrt{\sum_{x'} T(x',y')^2 \cdot \sum_{x'} I(x'+x,y')^2}} \quad (1)$$

We used ' ' as width pitch from single image.

2.1.2. Pitchwise Subtraction

On original image, pitchwise difference will be large on defect part, but small on background part. With this idea, we made middle_stage conversion image as M. Lets refer total gray image (all pixel value = 128) with same shape of original image a G, original image as O and pitchwise image shift function as S. M(x) which is pixel value of M is defined as formula (2) for all pixels on image.

$$M(x) = G(x) - \min[abs(O(x) - S_{Left}(O(x))), abs(O(x) - S_{Right}(O(x)))] \quad (2)$$

2.1.3. Make Binary Image

We set threshold between large and small difference of shift difference as 58, which means area with pixel value under 70 on M image, we consider there as defect part and over 70 as background part. binary image B is made by formula (3)

$$B(x) = \begin{matrix} 255 & if\ \{M(x) > 70\} \\ 0 & if\ \{M(x) \le 70\} \end{matrix} \quad (3)$$

2.2. Defect Feature Extraction

2.2.1. Find Blobs

Now in Binary image from 2.1.3, we have white pixels which indicate defects and black pixels which indicate background. We defined closed white part as blobs but, for blobs with area under 5 pixels, we considered as noise and converted into 255.

2.2.2. Feature Extraction

We extracted total 21 features of blobs found in 2.2.1 including 'Blob Area', 'Blob Brightness', 'Blobs Count', 'Maxblob Perimeter' and 'Maxblob Circularity', etc. We will call these features as 'Structured Defect Features' through this paper.

2.3. Convolutional Neural Network Classifier

We used VGG16 network [7] as baseline classifier. In addition, we adopted data augmentation [8], bottleneck structure [9], global average pooling (GAP) [10] and convolution with 2 strides in first convolution layer.

2.3.1. Cropped image

When training CNN, we tested both original image and cropped image around largest blob. Cropped image has 512x512 size with same center of maximal blob.

2.4. Wide and Deep Network

To combine structured data with image data, architecture of wide and deep network from Google Inc. [11] seems good for our problem. We concatenated structured features into output of global average pooling layer (which is input of fully connected layer) to combine two different shape of data. Figure3 below is format of full network architecture of wide and deep network with parameter settings.

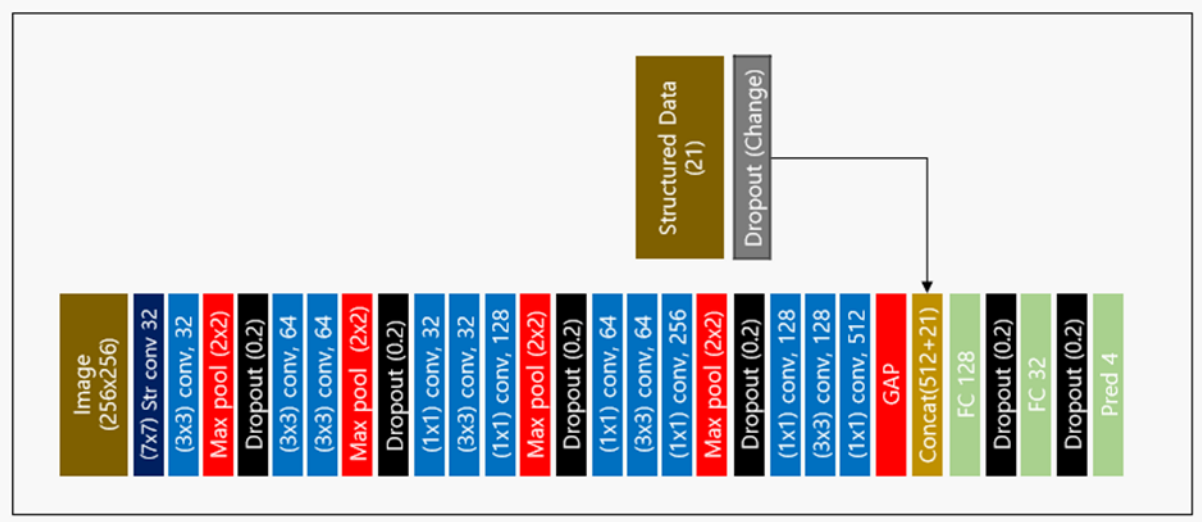

Fig. 3: Full Structure of Wide and Deep Model for Defect Classification

3. Results

We used Nesterov momentum 0.9, learning rate 0.0005, decay rate 0.00001 for Stochastic Gradient Optimizer, batch size 50, ReLU activation, and same padding. For image augmentation, zoom and flip option is used in keras.ImageDataGenerator, and step_per_epoch 120 (means total 6000 augmented image is used per 1 epoch). Classification accuracy of single classifier is represented in table1. We tested each model with GPU GTX 1080ti and takes about 6 hours to train each CNN models.

*MLP refers Multi Layer Perceptron.

*BE refers Background Eliminated images on 2.1.3.

TABLE I: ACCURACY OF SINGLE CLASSIFIER

Model	Structured Data (MLP)	CNN (original)	CNN (BE)	**CNN(BE_Crop)**
Test Accuracy	80.3%	74.8%	79.9%	**82.3%**

Background elimination and crop increases classification accuracy about 7.5% points. We tested wide and deep architecture changing dropout rate before concatenating structured data (gray layer on figure3) and other dropout layers' dropout rate is all same. We will refer these models as WD(n), n is the dropout rate on gray layer. Result of WD(n) models is on the table2.

TABLE II: ACCURACY OF WIDE AND DEEP ARCHITECTURE (BE_CROP)

Model	WD(0)	WD(0.4)	**WD(0.6)**
Test Accuracy	80.7%	85.3%	**86.6%**

Even though we combined structured feature, test accuracy of WD(0) model is worse than single CNN model and almost similar with single structured data model. But on WD(0.6) model, it outperforms single CNN model.

4. Conclusion and Future Works

We found that dropout rate on structured data input can make big difference of accuracy. It is because structured data plays many role on early training phase so gradients are concentrated to structured data part. This situation would disturb to train convolution part and it makes WD(0) model perform worse than a single CNN model. But with large dropout rate, it relieves gradient concentration on structured data part so it helps network to train both structured and image part with balance.

In other words, dropout can be a great technique to train data with different learning speed. And this technique leads wide and deep architecture works better for collaborating CNN features and human selected features. But we have no way to find the best dropout rate of each input but just make a lot of experiments. Also, if we load very large dropout rate, total training time become much longer than single models. So if we find generalized techniques to collaborate multiple inputs, we can get better classification result for images on specific domain.

5. Acknowledgements

This work was supported by the BK21 Plus (Big Data in Manufacturing and Logistics Systems, Korea University) and aim Systems.

6. References

[1] Yixiang F. Zhang and Randall R. Bresee “Fabric Defect Detection and Classification Using Image Analysis” Textile Res. J. 65(1), 1-9 (1995)

https://doi.org/10.1177/004051759506500101

[2] K. Kameyama, Y. Kosugi, T. Okahashi, and M. Izumita “Automatic Defect Classification in Visual Inspection of Semiconductors Using Neural Networks” IEICE TRANS. INF. & SYST., VOL. E81-D, NO.11 November 1998

[3] T. Y. Lim, M. M. Ratnam and M. A. Khalid “Automatic classification of weld defects using simulated data and MLP neural network” Insight Vol 49 No 3 March 2007

https://doi.org/10.1784/insi.2007.49.3.154

[4] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber, “Steel Defect Classification with Max-Pooling Convolutional Neural Networks,” WCCI 2012 IEEE World Congress on Computational Intelligence Jun. 10-15, 2012 - Brisbane, Australia

https://doi.org/10.1109/IJCNN.2012.6252468

[5] T. Wang, Y. Chen, M. Qiao, and H.Snoussi, “A fast and robust convolutional neural network-based defect detection model in product quality control” Int. J. Adv. Manuf. & Technol. DOI 10.1007/s00170-017-0882-0

https://doi.org/10.1007/s00170-017-0882-0

[6] Y. Lecun, L. Bottou, Y. Bengio and P.Haffner, “Gradient-Based Learning Applied Document Recognition” Proc of The IEEE, Nov. 1998

https://doi.org/10.1109/5.726791

[7] K. Simonyan and, A. Zisserman “Very Deep Convolutional Networks for Large-Scale Image Recognition” arXiv:1409.1556v6 [cs.CV] 10 Apr. 2015

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks” Advances in Neural Information Processing Systems 25 (NIPS 2012)

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition” arXiv:1512.03385v1 [cs.CV] 10 Dec. 2015

[10] M. Lin, Q. Chen, and S. Yan “Network in Network” arXiv:1312.4400v3 [cs.NE] 4 Mar. 2014

[11] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T, Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, “Wide & Deep Learning for Recommender Systems” arXiv:1606.07792v1 [cs.LG] 24 Jun. 2016