

# Detecting Defects in PCB using Deep Learning via Convolution Neural Networks

Venkat Anil Adibhatla  
Dept. Mechanical Engineering and  
Innovation Center for Big Data  
and Digital Convergence, Yuan Ze  
University.  
Chung-Li, Taiwan  
[venkataniladibhatla@gmail.com](mailto:venkataniladibhatla@gmail.com)

Jiann-Shing Shieh  
Dept. Mechanical Engineering and  
Innovation Center for Big Data  
and Digital Convergence, Yuan Ze  
University.  
Chung-Li, Taiwan  
[jsshieh@saturn.yzu.edu.tw](mailto:jsshieh@saturn.yzu.edu.tw)

Maysam F. Abbod  
Dept. Electronic and Computer  
Engineering  
Brunel University London  
Uxbridge, UK  
[maysam.abbod@brunel.ac.uk](mailto:maysam.abbod@brunel.ac.uk)

Huan-Chuang Chih  
Dept. Advanced manufacturing  
system, Boardtek electronics  
corporation.  
Chung-Li, Taiwan  
[strong@mail.boardtek.com.tw](mailto:strong@mail.boardtek.com.tw)

Chi-Chang Hsu  
Dept. Advanced Research &  
System Develop Division,  
Boardtek electronics corporation.  
Chung-Li, Taiwan  
[chang@mail.boardtek.com.tw](mailto:chang@mail.boardtek.com.tw)

Joseph Cheng  
President, Boardtek electronics  
corporation.  
Chung-Li, Taiwan  
[JosephCheng@mail.boardtek.com.tw](mailto:JosephCheng@mail.boardtek.com.tw)

**Abstract**—In this paper we have deployed the concept of deep learning known as convolutional neural networks (CNN) as we can realize nowadays deep learning is growing in each and every field. Deep learning is executed in each and every platform and its outcome is impressive. On the other hand, the capability and accuracy of deep learning is somehow compared with human beings. We trained CNN to classify either defective or good printed circuit board(PCB). In this experiment we have used 41,387 images, which is divided into 3 different data sets i.e. training, validation and testing. The CNN, which has 60 million parameters and 500,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and two globally connected layers with a final 1000-way softmax. Hence, deep learning via convolution neural networks has been introduced in this paper, which will eventually increase the accuracy and reduce a lot of time and consumption of skilled manpower. According to this preliminary study, we can overall achieve accuracy of above 88% and minimize the count of defective PCB classifying as good. In the near future, we hope that over 95% accuracy can be achieved by using different CNN models like VGGNET, RESNET and GOOGLNET and collecting more PCB image data in order to reduce the consumption of time, manpower and increase the accuracy in quality inspection.

**Keywords**—Convolution neural network, Deep learning, LeNet, ALEXNET, GOOGLNET, Printed circuit board, Good class, Damaged class, Confused class, Convolution layer, Max pooling, fully connected layers.

## I. INTRODUCTION

The Printed Circuit Board (PCB) is very important in all electronic gadgets, which are used either for domestic use or for industrial purpose. These are thin plates which are usually made of fiberglass, laminated materials, or composite epoxy used as a physical base to support chips and electronic components [1,2,3,4]. These boards are printed or etched with conductive pathways to form circuits that will power electronic devices or gadgets. They support different electronic components like integrated circuits, resistors, and

transistors, which are tie-in with copper tracks. These components are mounted by drilling holes in the board and soldering them into the circuit pattern. During this process, the PCB can be damaged and it has to be detected before it explores the market. The importance of the inspection process has been enhancing by the requirements of the modern manufacturing environment. In electronics mass-production manufacturing facilities, an attempt is often made to achieve 100% quality assurance of all parts, subassemblies, and finished goods. A variety of approaches for automated visual inspection of printed circuits have been reported over the past two decades. Detection of bare PCB defects by image subtraction method using machine vision method [5,6], in this project we have deployed the concept of deep learning known as convolution neural network as we can realize nowadays deep learning is growing in each and every field. Especially, the deep learning is introduced to the production line to reduce the manpower consumption and increase the accuracy in detecting defects of PCB.

## II. PURPOSE

As we know PCB is the most important and crucial part of each and every electronic component and now days' electronic product are used in each and every product in our day to day life [7]. Hence, our life is reliable to the electronics. Any defect or malfunction in PCB may lead to catastrophic condition to the human beings so the industries who manufacture the PCB need to be sure that the defective or damaged PCB should be detected and restrict it from going into hand to hand use. There are many image subtraction techniques that has been implemented in electronics industries [8,9] in the past but as we know deep learning has been a booming concept now days in detecting and classifying objects with an appreciable accuracy [10,11]. Detecting and classifying the good PCB and defective PCB would avoid many accidents and save the time of human beings.

### III. CONVOLUTION NEURAL NETWORK

There are five main operations in the CNN input layer, convolutional layer, Relu layer, pooling layer and fully connected layer. In addition to it in Figure 1, the structure of CNN is displayed with different layers.

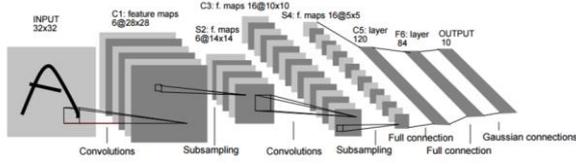


Fig. 1. Structure of convolution neural network [12]

#### A. Input Layer

Input [32x32x3] will hold the raw pixel values of the image. In this case, an image of width 32, height 32, and with three color channels red, green, blue (R, G, B) is shown in input of Figure 1.

#### B. Convolutional Layer

The convolutional layer is the core building block of a convolutional network. Let's first discuss what the convolution layer computes. The convolution layer's parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. For example, a typical filter on the first layer of a convolutional networks (ConvNet) might have size 5x5x3 (i.e. 5 pixels' width and height, and 3 because images have depth 3, the color channels). During the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. As we slide the filter over the width and height of the input volume we will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual features such as an edge of some orientation or a blotch of some color on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network. Now, we will have an entire set of filters in each convolution layer and each of them will produce a separate 2-dimensional activation map. We will stack these activation maps along the depth dimension and produce the output volume.

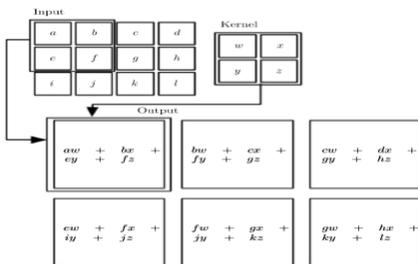


Fig. 2. Working of 2 Dimensional convolutions [13]

An example of 2 dimensional convolutions is shown in Figure 2. In this case, we restrict the output to only positions where the filter or kernel lies entirely within the image. We draw boxes with arrows to show how the upper-left element of the output is created by applying the filter to the corresponding upper-left region of the input. If we use a two-dimensional image  $i$  as our input, we probably also want to

use a two-dimensional filter (i.e.,  $k$ ) and we can also write as follows.

$$S(i,j)=(k*i)(i,j)= \sum m \sum n I(I - m,j - n)k(m,n) \quad (1)$$

#### C. ReLU Layer

An additional operation called ReLU has been used after every convolution operation in Figure 3. ReLU stands for rectified linear unit and is a non-linear operation.

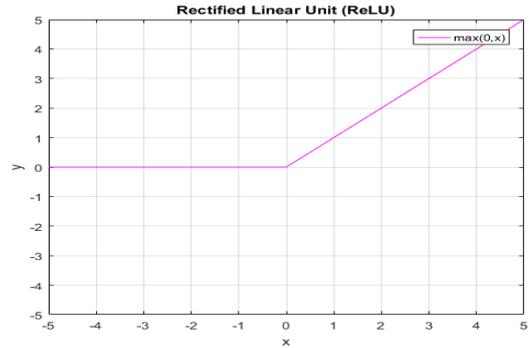


Fig. 3. Graphical representation of ReLU [13]

ReLU layer will apply an elementwise activation function, such as the  $\max(0, x)$  threshold holding at zero. ReLU is an element-wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero. The purpose of ReLU is to introduce non-linearity in our ConvNet since most of the real-world data we would want our ConvNet to learn for non-linear (Convolution is a linear operation element wise matrix multiplication and addition, so we account for non-linearity by introducing a nonlinear function like ReLU). This leaves the size of the volume unchanged.

#### D. Pooling Layer

The pooling or down sampling layer is responsible for reducing the spacial size of the activation maps. In general, they are used after multiple stages of other layers (i.e. convolutional and non-linearity layers) in order to reduce the computational requirements progressively through the network as well as minimizing the likelihood of overfitting. Spatial pooling (also called subsampling or down sampling) reduces the dimensionality of each feature map but retains the most important information. Spatial pooling can be of different types: Max, Average, Sum etc.

In case of max pooling, we define a spatial neighborhood (for example, a 2x2 window) and take the largest element from the rectified feature map within that window. Instead of taking the largest element we could also take the average (average pooling) or a sum of all elements in that window. In practice, Max Pooling has been shown to work better.

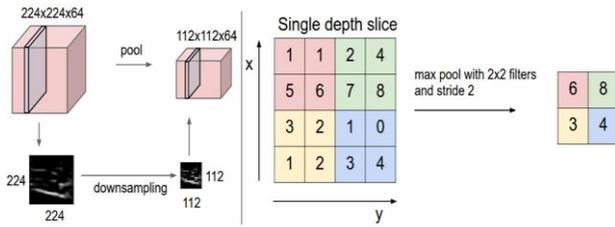


Fig. 4. Pooling layer down samples the volume spatially, independently in each depth slice of the input volume.

We slide our 2 x 2 window by 2 cells (also called ‘stride’) and take the maximum value in each region. As shown in Figure 4, this reduces the dimensionality of our feature map. The function of pooling is to progressively reduce the spatial size of the input representation. In particular pooling, it makes the input representations (feature dimension) smaller and more manageable. Reducing the number of parameters and computations in the network, controlling overfitting makes the network invariant to small transformations. Distortions and translations in the input image of a small distortion in the input will not change the output of pooling since we take the maximum/average value in a local neighborhood. It helps us arrive at an almost scale-invariant representation of our image. This is very powerful since we can detect objects in an image no matter where they are located.

#### E. Fully Connected Layer

Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular neural networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset. The output from the convolutional layers represents high-level features in the data. While that output could be flattened and connected to the output layer, adding a fully-connected layer is a cheap way of learning nonlinear combinations of these features.

#### F. Output

After connecting all the previous layer with the help of fully connected layer, it’s time for classifying the output. I would explain how does it predict the output with a small example. Let us consider we have trained a set of images which has 2 classes X and O after the training such 2 classes. We provide the input i.e. to classify whether it is X or O. As in the Figure 5, we can see that the weights represent that there is 92% probability that the given input is classified as X and 51% is O.

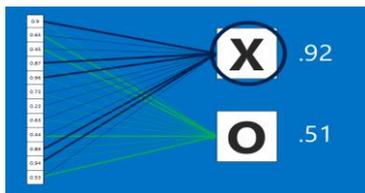


Fig. 5. Output prediction

After having the probability percentage, we need to know how does it calculate these percentages and from where does these magic numbers come from.

TABLE I. CALCULATION OF ERROR

	<i>Right answer</i>	<i>Actual answer</i>	<i>Error</i>
X	1	0.92	0.08
O	0	0.51	0.49

In CNN for 2 classes, we use binary cross entropy and the function known as sigmoid during the application of this function. Our result will be displayed in 0 or 1. As we can see in the Table I, X is represented as 1 and 0 is represented as 0. It also displays that there is a 92% probability of X as an input and 51% probability of 0. The table also represents a 57% of total error displaying 8 % for X and 49% for O. Depending on these trained weights, the CNN detects and classifies the images on the ratio of probability.

#### IV. METHODOLOGY OF EXPERIMENT

Recently, couple of months before I have visited the PCB industries and precisely gone through the production and quality inspection line in the industry, we realized that a lot of skilled manpower and time is dedicated in quality inspection of the product to deliver a product that is 100% perfect. But this whole process can also be done with the help of CNN (convolution neural network), which would eventually reduce a lot of time, manpower and increase the accuracy. To implement this method, Images of PCB has been extracted for a preliminary study. Almost 47428 images have been used in this experiment which is divided into 3 classes good, confused and damaged. It would have been little strange when I mention confused class. This class is actually a good class but it looks like damaged which means it has some small scratch or external dirt attached to it but it can be used for the further process. Let me display the sample images of all the 3 classes. Figures 7, 8, and 9 display sample image for good, confused and damaged classes.

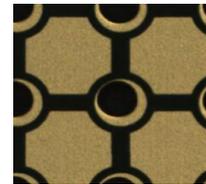


Fig. 7. Sample image of good class



Fig. 8. Sample image of confused class

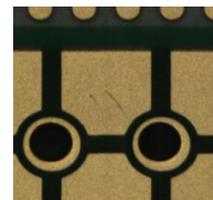


Fig. 9. Sample image of damaged class

This preliminary study has been initiated using 2 stages classification process. In the first stage, the confused and damaged images are addressed to the single class and good images to another class with the help of flowchart. We have

shown how does the classification process work as shown in the Figure 10. We have used 47,428 images for 1st stage. 34,147 set of images is being used for training in which 10,353 images belongs to the confused class and 23,794 to good class, 8538 set of images are used for validation in which 2589 belongs to confused and 5949 belong to good. After training the CNN structure 4743 images are used to test the 1st stage CNN.

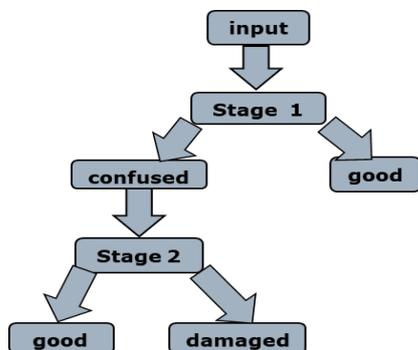


Fig. 10. Flow process of experiment

Apart from these images, another CNN has been trained which is 2nd stage training at this stage we have distributed the set of images there are 14380 images used for 2nd stage. 10353 set of images is being used for training and 2589 images are used for validation. After training and validation, the CNN structure of 1438 images are used to test the 2nd stage of CNN.

## V. RESULTS

After training we have tested it with a new set of data, below in Tables II and III, we had displayed the result of 5 cross validations for stage 1 and another 5 cross validations for the 2nd stage which displays 10 cross-validations.

Table II. MEAN AND SD FOR 1<sup>ST</sup> STAGE TRAINING, VALIDATION AND TESTING

	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
Crossvalidation1	98.67%	96.74%	87.26%
Crossvalidation2	98.76%	96.40%	89.22%
Crossvalidation3	98.65%	96.56%	89.62%
Crossvalidation4	98.56%	96.64%	87.68%
Crossvalidation5	98.93%	96.28%	87.03%
Mean±SD	98.71±0.140	96.52±0.184	88.16±1.180

Table III. MEAN AND SD FOR 2<sup>ND</sup> STAGE TRAINING, VALIDATION AND TESTING

	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
Crossvalidation1	92.90%	87.99%	81.00%
Crossvalidation2	92.93%	86.95%	84.02%
Crossvalidation3	92.89%	87.92%	83.03%
Crossvalidation4	92.92%	88.02%	81.56%
Crossvalidation5	92.87%	88.76%	84.56%
Mean±SD	92.90±0.021	87.92±0.576	82.83± 1.371

As we can see the result of classification is really appreciable, and we are able to classify the good and damaged PCB with a consistent accuracy of 85% for both the

stage. For any kind of machine learning application, large dataset and the algorithm play the vital role as we can see result of classifying. The good PCB were better than classifying defective one due to having dataset of good PCB more than defective one.

## VI CONCLUSION

This paper proposed a new highly accurate defect classification method without using reference images by CNN feature extraction. The effectiveness of the proposed method was validated by an experiment for detecting the defect using actual images of electronic circuit boards. Defect detection without reference images was implemented by performing the proposed method. As future work, there is higher accuracy of detection and classification.

## ACKNOWLEDGMENT

This research was supported by the two centers Lenovo Harold lab and Boardtek electronics corporation.

## REFERENCES

- [1] Suzuki H, inventor; Junkosha Co Ltd, assignee, 1987. Printed circuit board. United States patent US 4,640,866.
- [2] Matsubara H, Itai M, Kimura K, inventors; NGK Spark Plug Co Ltd, assignee, 2003 Printed circuit board. United States patent US 6,573,458.
- [3] Magera JA, Dunn GJ, inventors; Motorola Solutions Inc, assignee, 2008. Printed circuit board. United States patent US 7,459,202.
- [4] Cho HS, Yoo JG, Kim JS, Kim SH, inventors; Samsung Electro-Mechanics Co Ltd, assignee, 2012. Printed circuit board. United States patent US 8,159,824.
- [5] Chauhan, A.P.S. and Bhardwaj, S.C., 2011. Detection of bare PCB defects by image subtraction method using machine vision. In Proceedings of the World Congress on Engineering (Vol. 2, pp. 6-8).
- [6] Khalid, N.K. and Ibrahim, Z., 2007. An image processing approach towards classification of defects on printed circuit board (Doctoral dissertation, University Technology Malaysia).
- [7] Bauder R, inventor; Reitter, Schefenacker GmbH and Co KG, assignee, 1999. Taillight fixture for motor vehicles with printed circuit boards with connectors and LED's. United States patent US 5,975,715.
- [8] Wu, W.Y., Wang, M.J.J. and Liu, C.M., 1996. Automated inspection of printed circuit boards through machine vision. Computers in industry, 28(2), pp.103-111.
- [9] Moganti, M. and Ercal, F., 1995. Automatic PCB inspection systems. IEEE Potentials, 14(3), pp.6-10.
- [10] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C. and Xu, W., 2016, June. Cnn-rnn: A unified framework for multi-label image classification. In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on (pp. 2285-2294). IEEE.
- [11] Socher, R., Huval, B., Bath, B., Manning, C.D. and Ng, A.Y., 2012. Convolutional-recursive deep learning for 3d object classification. In Advances in Neural Information Processing Systems (pp. 656-664).
- [12] LeCun, Y., Bottou L., Bengio Y., and Haffner P., 1998. Gradient-Based Learning Applied to Document Recognition", Proc. Of the IEEE, Nov., pp 1-46.
- [13] Goodfellow, I. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.