Detection of Printed Circuit Board Defects on ENIG and ENIPIG Surface Finishes with Convolutional Neural Networks and Evaluation of Training Parameters

Angelika Hable,1* Paul Tabatabai,1 Hannah L. Lichtenegger,1 Anton Scherr,2 Thomas Krivec,3 and Dieter P. Gruber1

Abstract—Increasingly high demands are being placed on the quality inspection of printed circuit boards (PCBs). A full surface inspection of all produced PCBs and a high defect detection accuracy of the inspection system are becoming prerequisites for an efficient quality management. At the same time, the demand for PCBs is constantly increasing over the years due to the high demand for electrical devices. Human inspection is no longer feasible due to the high production rates and required defect detection accuracy. Therefore, automatic inspection systems are increasingly used for quality control in the various process steps of PCB production. In this article, the first automatic inspection system for detecting defects on Electroless Nickel Immersion Gold (ENIG) and Electroless Nickel Immersion Palladium Immersion Gold (ENIPIG) surfaces is presented. A pretrained convolutional neural network (CNN) and the sliding window approach are used. A training dataset consisting of six different defect types and an OK class containing only defect-free PCB images was labeled for this classification problem. The hyperparameters learning rate and batch size are varied for different training runs of the CNN, and the performance of the network in PCB defect detection is evaluated using a test dataset. The true-positive rate, true-negative rate, and F1-score were analyzed for the evaluation. Our results show that the best performances could be achieved at low batch sizes and low learning rates.

Keywords—Printed circuit boards, defect detection, quality inspection, neural networks, PCB surface finishes

INTRODUCTION

Printed circuit boards (PCBs) are included in most electronic devices, such as computers, cell phones, and tablets. The demand for electronic devices is constantly increasing, so the proportion of PCBs produced is also continuing to rise. At the same time, the demands on quality inspection are increasing. All PCBs must be inspected and a high defect detection accuracy is expected. This is no longer feasible by human optical visual inspection. Automatic inspection systems provide a fast and objective evaluation of defects and are adapted to the respective PCB production step and the defects occurring at that step. The production of PCBs comprises more than 50 process steps [1]. There are inspection methods, which deal with the detection of defects on circuit patterns [2-4], surface finishes [5-7], flux defects [8], and the placement of components [9-10] on the PCBs. These methods can be divided into two categories: template comparison methods and non-comparative methods. Template comparison methods compare the PCB to be inspected with a defect-free PCB, the template. In the process, either image captures or features of the PCBs are used. The features are extracted from the images of the PCBs using convolutional backbones [2], autoencoders [9], or genetic algorithms [11]. Comparison methods compare images/feature using subtraction [12], exclusive disjunction methods [13-14], normalized cross-correlation [11], or similarity score [9]. Noncomparative procedures do not use the layouts and features of good parts. These procedures can be divided into filter methods [6, 7, 15], mathematical methods [5], and methods with neural networks [3, 4, 10]. Filters and mathematical methods are applied to the image of the captured PCB. As a result, it is possible to distinguish between background and defect. Convolutional neural networks (CNNs) can be used for defect detection and classification.

In this article, we deal with the detection and classification of defects on surface finish. Thereby, Electroless Nickel Immersion Gold (ENIG) and Electroless Nickel Immersion Palladium Immersion Gold (ENIPIG) platings were investigated. ENIG consists of a nickel layer covered by a thin layer of gold [16]. ENIPIG has an additional layer of palladium between the gold and nickel layer [17]. The purpose of surface finishing is to protect the copper from corrosion before electronic components are soldered onto the copper. Corrosion can impair the soldering of the components and thus lead to a circuit board rejection. To prevent excessive rejects and to avoid wasting resources, it is important to ensure that the gold coating is not affected by defects and that the copper is protected from environmental influences. The quality inspection of these gold areas is, therefore, an important part of the manufacturing process of PCBs. Defects on the gold areas include copper residues, mechanical damages, and discolorations.

There are only few inspection algorithms that have been developed for the detection of defects on the gold areas [5-7].
Wang et al. [7] used two-dimensional (2D) wavelet packet frame decomposition for the detection of contamination, discoloration, scratches, and bare copper parts on gold areas of PCBs. The gray scale image of the PCB under test is decomposed into several subbands. Subsequently, the subwindow average energies are compared and corresponding features are extracted from important subbands. With the help of Principal Component Analysis and thresholding methods, a defect mask is created. Xie et al. [5] developed a mathematical model that can detect defects, such as discoloration or scratches, on gold areas by identifying local color differences between defect and background. For each image pixel, a statistical value (weighted sum of a color deviation term and a color fluctuation term) is calculated, which indicates whether the pixel should be counted as a defect or not. This way, each image pixel can be classified as defect-free or defective. Tsai et al. [6] examined gold-plated edge fingers of PCBs regarding four different defects: copper exposure, pinholes, rough surface, and nicks. The defects can be detected with two specially developed entropy measures that can detect color changes and directional irregularities. The studies mentioned earlier use image processing and mathematical methods to detect the defects on gold surfaces. With these methods only, a detection of the defects is possible and not a classification. The parameters used in these methods are strongly adapted to the existing defect data so that deviations, in the appearance of the defects and gold surface or by changed environmental influences as, e.g., by light adjustment, can lead to an incorrect defect detection. By using CNNs, this source of error is considerably reduced and, in addition to the detection of defects, a classification of these is also possible. We want to stimulate research in this important area and present a high-resolution inspection system for the detection and classification of defects on gold areas of PCBs. The defect evaluation presented in this article is performed using a pretrained CNN. The evaluation on a test dataset considers the true-positive rate (TPR), the true-negative rate (TNR), and the F1-score at different batch sizes and learning rates.

This article is structured as follows: Section 2 Describes the PCBs used, the defects to be detected, and the hardware components. In Section 3, a description of the defect detection with CNNs is given. In Section 4, the inspection system is benchmarked and results are presented and discussed.

PCB SAMPLES, PLATING DEFECTS, AND INSPECTION HARDWARE

In this article, two different PCB types with gold plating were investigated. Sample 1 has an ENIPIG plating while sample 2 has an ENIG surface finish. The PCBs have a diameter of 17 mm (sample 1) and 20 mm (sample 2) and are arranged in an array structure. Array structures of small PCBs were chosen to be the subject of this study because this layout was most relevant for our project partner. The samples have 73 (sample 1) and 48 (sample 2) different gold areas with different geometric shapes, such as polygons and circles. In sample 1, the smallest gold pad is 0.430 × 0.285 mm and the largest is 3.450 × 0.850 mm. In sample 2, the smallest gold pad is 0.2 × 0.3 mm and the largest is 3.0 × 1.5 mm. When inspecting the gold surfaces, all gold areas were inspected with the same detection accuracy. Despite different surface finish, the same defect types are found on PCB samples. The defects are between 20 and 500 μm in size and contrast differently to the gold background. A selection of occurring defects is shown in Fig. 1.

1. Incompletely filled vias: Vias are electrical connections among circuit pattern levels of a PCB. These holes are filled with plug-in paste. If the vias are not completely filled, no flat surface is formed after nickel and gold plating.
2. Copper residue: Remains of the copper layer that could not be removed during the cleaning process. They visually appear as mounds on the ENIG or ENIPIG surfaces.
3. Dent: Dents are caused when residues from removed particles leave pressure marks in the copper layer.
4. Mechanical damage: Mechanical damage is caused by improper handling during production. One example of mechanical damage is scratches on the gold surface.
5. Discoloration: Discoloration of the gold layers occurs, e.g., due to fingerprints or other residues in the production process.
6. Contamination: Contamination particles on the gold layer such as threads or other dust particles can get onto the circuit boards after gold deposition.

The inspection system uses a color area scan camera, which has a resolution of 28.8 megapixels. A telecentric lens from SILL Optics with a magnification factor of 1 is used. Using this setup, a camera frame with the dimension of 36.2 mm × 24.1 mm can be captured with a resolution of 5.5 μm per pixel. For the defect detection, sufficient contrast and brightness must be available [18]. For this reason, the described setup uses a coaxial incident light, enabling the homogeneous and diffuse illumination of the PCBs. Fig. 2 shows the camera setup and the coaxial incident light illumination.

The inspection system is designed to detect particularly small defects with high resolution and high speed. This means that the camera sensor cannot capture the whole PCB array at
once, but—using an XY-stage—captures one PCB after the other. The XY-stage, which is integrated into the inspection system, moves the PCBs under the camera during the inspection for image acquisition. Fig. 3 shows the whole inspection system.

**CONVOLUTIONAL NEURAL NETWORKS FOR DEFECT DETECTION AND CLASSIFICATION**

CNNs consist of several convolution layers that extract simple features from the input image in the lower layers and increasingly complex features in the higher layers. At the end, there are one or more fully connected layers that use these features to output classification results. The convolutional and fully connected layers of a CNN have tunable parameters, called weights. The functionality and performance of a neural network are only determined by its architecture and learned weights. Which features are extracted by the convolutional layers depends on the learned weights. At the beginning of the training process, the weights are randomly initialized or if a pretrained network is used, the weights are initialized with the pretrained values. During the training process, the weights are continuously adjusted so that a chosen loss function is minimized. The loss function indicates the error between the prediction of the neural network and the real value of the dataset. In the case of classification problems, usually the cross-entropy loss,

\[
L(\hat{y}) = - \sum_{i=0}^{n} y_i \log(\hat{y}_i) \tag{1}
\]

is used. It gives information about the class prediction accuracy. In equation (1), \(n\) is the number of classes, \(y\) is the vector of class labels, and \(\hat{y}\) is the vector of predicted class labels. During training, the CNN sequentially receives the training images, grouped together in batches as input. The batch size specifies how many training images are contained in a group. After a group of images (a batch \(B\)) passes through the network, the cross-entropy loss (of these images) is calculated and the weights are updated. If \(w\) denotes the vector of all weights of a CNN, the simplest possible weight update is described by

\[
w_{\text{new}} = w - \alpha \cdot \frac{\partial}{\partial w} \left( \frac{1}{|B|} \sum_{b \in B} L(\hat{y}_b) \right), \tag{2}
\]

where, \(\alpha\) is the learning rate parameter. It specifies in which order of magnitude the weights are adjusted. Parameter \(\frac{1}{|B|} \sum_{b \in B} L(\hat{y}_b)\) is the average loss over the predictions \(\hat{y}_b\) of one batch \(B\). After adjusting the weights, another image batch runs through the network and the loss is calculated again and the weights are updated. These steps are repeated until all training images have been run through the network. An iteration of the entire training images through a CNN is called an epoch. CNNs are trained over multiple epochs.

Suitable training parameters (e.g., batch size and learning rate) must be selected for optimal defect detection. Which values are suitable depends on the network architecture and the dataset. Metrics that measure the classification performance are used to evaluate the defect detection. For this purpose, the CNN is used to classify image data from PCBs that are not included in the training data and are thus unknown to the CNN. In this article, we use the TPR, the TNR, and the \(F_1\)-score to measure classification performance. These are binary metrics. This means they provide information on how well the CNN can distinguish between defects and nondefects (OK).
The TPR [23]

\[
TPR = \frac{TP}{TP + FN},
\]  

(3)
describes the proportion of defects correctly detected by the neural network. It is also called sensitivity or recall. In equation (3), TP indicates the number of true positives. This is the number of images to which the CNN correctly assigned the defect class. The number of the defect images that were incorrectly assigned the OK class is called false negative (FN). A high TPR value means that a large number of defects are found. However, OK images will also be classified as a defect.

The TNR

\[
TNR = \frac{TN}{TN + FP},
\]  

(4)
describes the proportion of OK images classified as correct and is also called specificity. In equation (4), TN indicates the number of true negatives. This is the number of OK images that were correctly classified by the CNN. The number of false positives (FP) indicates the number of OK images that were incorrectly assigned the defect class. The TPR and the TNR are correlated, which means that a high TPR usually means a low TNR and vice versa.

The precision [24]

\[
\text{precision} = \frac{TP}{TP + FP},
\]  

(5)
describes the proportion of correctly classified images. A high precision value means that the detected defects are very likely to be correctly detected defects, but this value does not provide any information about the number of unrecognized defects.

The F1-score [23]

\[
F1 - \text{score} = 2 \cdot \frac{\text{Precision} \cdot TPR}{\text{Precision} + TPR},
\]  

(6)
is a measure of CNN performance and is calculated from precision and TPR equally weighted. A high F1-score means that the precision and TPR values are high. In other words, a large proportion of the defects present on a PCB are detected with a high probability that they are real defects.

The CNN architecture ResNet18 [25] is used for the detection of PCB defects. To use this architecture, the last layer of this network, a fully connected layer, is replaced by a dense layer with seven output neurons for the number of defects classification. The PCBs used were produced over a dimension of approximately 1 y and, therefore, originate from different production runs. Additionally, a number of defect-free OK regions were labeled. In the labeling process, image patches of size 100 px × 100 px containing the defects and OK regions were extracted from the PCB images. Data augmentation is used to artificially increase the amount of training data and also prevents the neural network from overfitting on the training set [23]. In this technique, the training image patches are subjected to random image transformations, changing the image while preserving the semantic information contained in it. The image transformations used in the presented experiment are affine transformations (translation and rotation), shear as well as brightness and contrast changes and reflections. Each labeled defect was extracted multiple times, each time after a new random augmentation of the respective PCB image. Table I represents the fraction of defect classes in the total training dataset composed of augmented image patches. Some defects are rare and result in the dataset being unbalanced.

<table>
<thead>
<tr>
<th>Defect type</th>
<th>Number of images for training the CNN</th>
<th>Dataset ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copper residue</td>
<td>18,837</td>
<td>14.90</td>
</tr>
<tr>
<td>Mechanical damage</td>
<td>5,427</td>
<td>4.29</td>
</tr>
<tr>
<td>Dent</td>
<td>25,589</td>
<td>20.25</td>
</tr>
<tr>
<td>Incompletely filled vias</td>
<td>759</td>
<td>0.60</td>
</tr>
<tr>
<td>OK</td>
<td>40,882</td>
<td>32.35</td>
</tr>
<tr>
<td>Contamination</td>
<td>18,759</td>
<td>14.84</td>
</tr>
<tr>
<td>Discoloration</td>
<td>16,136</td>
<td>12.77</td>
</tr>
</tbody>
</table>

Distribution of the Training Dataset
During inference, the sliding window approach is used to evaluate the PCB images with the CNN. It is a common method for localization of objects in images [31]. The CNN assigns a defect class or OK class to the input image. The sliding window approach is used to segment the image frame of the PCB into several smaller image patches, which subsequently serve as input to the CNN. Thus, each patch is classified individually. This method offers the advantage that the defect can be localized in the positional accuracy of the image patch and different defect classes can be detected on a PCB. In the sliding window method, a window with a certain dimension (in our case 100 px × 100 px) is sliding over the entire image to create image patches to be processed by the CNN.

RESULTS AND DISCUSSION

The ResNet18 was trained for eight epochs, varying batch sizes (4, 8, 16, 32, 64, 128, and 1,024) and learning rates $10^{-3}$ and $10^{-4}$. Batch sizes >1,024, and some other candidate values for the learning rate parameter, were also tested, but discarded due to not performing as well as the mentioned values. For training, the cycle policy described in Section 3 was used. One neural network is trained per batch size and learning rate. The test dataset consists of a total of 150 PCB images (3050 px × 3050 px size) of sample 2. The PCBs for testing were, in contrast to the training dataset, produced close together in time. Therefore, it might be possible that the PCBs for testing share some special characteristics related to the production process, but we suspect that it had no relevant effect. The defect class incompletely filled vias is not present in the test dataset, therefore, we do not evaluate the networks’ ability of predicting this defect class. The evaluation of the circuit board images is carried out via the sliding window approach. Fig. 4 shows a section of an evaluated PCB image. The evaluation of the networks on the test dataset only distinguishes among defect patches, a patch to whom any defect class has been assigned, and good patches, a patch to whom the OK class has been assigned. This way, there is no examination of the correct defect classification by the CNNs, but an investigation of the general defect detection capability. In the evaluation, any patch that is assigned a defect class is counted as a defect. PCB defects marked by multiple patches will, thus, be counted multiple times. To calculate the TPR, TNR, and F1-score, the number of correct defect patches (TP) and correct OK patches (TN) were counted for each image from the evaluated test dataset. In the same way, the number of FNs and the number of FP were determined. This calculation was performed for both learning rates and eight different batch sizes. To calculate the average number of correct detections per defect, the sum of all defects correctly detected by the CNN (a defect can be counted more than once) is divided by the sum of detected defects (a defect can only be counted once).

Figs. 5 (a), (b) and 6 (a), (b) show the TPR and TNR of the conducted training experiments. The graphs show that the TPR...
increases with higher batch size up to a maximum determined by the learning rate and then decreases again. The TNR behaves in the opposite way, with increasing batch size, the TNR falls to a minimum and rises again with higher batch size.

Due to the sliding window approach, a single defect can be detected multiple times. Fig. 7 (a), (b) give the average number of defect detections (patches classified as defective) per defect for varying batch sizes.

Interestingly, the behavior of the average number of defect detections per defect does not exactly follow the behavior of the TPR: For learning rate $10^{-3}$ and batch size 256, the average number of defect detections per defect is less than the one for batch size 128 while the TPR increases with increasing batch sizes up to 512. Fig. 8 (a), (b) show the $F_1$-score for the learning rates $10^{-3}$ and $10^{-4}$ for the examined batch sizes.

The maximum $F_1$-score for learning rate $10^{-3}$ is at batch size 64 and has a value of 93.6%. For learning rate $10^{-4}$, the $F_1$-score has the maximum at batch size 16 with the result of 93.9%. The maxima of the $F_1$-score values are not at the same batch sizes as the maxima of the TPRs and the minima of the TNRs, which is explained by the influence of the precision as it falls with higher batch size. The best results of the $F_1$-score reflecting the performance of classification were obtained at learning rate $10^{-4}$ and batch size 16. These results show that the best evaluation results are achieved at lower batch sizes and learning rates, even though for this dataset, the results at different learning rates are close to each other.

In general, the loss function converges to a flat minimizer when training with a small batch size, as opposed to a sharp minimizer when training with a large batch size [32]. Flat minimizers generalize better [32], which is consistent with our experiments, where small batch sizes performed best on the test dataset. Our results are consistent with Kandel et al. [33], who observed that small batch sizes and small learning rates provided the best performance of CNNs. However, in comparison, Kandel et al. [33], who trained their networks with medical image dataset and achieved the best performance at batch sizes 16, 32, and 256, it becomes clear that optimal parameters...
CONCLUSIONS

In this article, we present the first optical inspection system for PCBs for the detection of defects on ENIG and ENIPIG surfaces. The inspection system was developed within a research project and was successfully tested in an industrial environment. We have estimated that a human inspector can examine about 600-1,200 PCBs per hour. With our system, we have an estimated 1,800 PCBs per hour. Using a pretrained ResNet18 CNN, six different surface defects can be detected. A dataset consisting of the defect classes to be detected was generated and the sliding window approach is used to localize and classify the defects. A comparison of the detection accuracy with [5-7], who have also developed methods for the detection of defects on gold surfaces, is difficult because in each case, own datasets were used. An advantage of our method over the others is that in addition to defect detection, a defect classification is also performed. Furthermore, our defect detection method can be used to examine inspection areas with incompletely filled gold areas that do not contain a complete gold surface. This is not possible with the mathematical model used by Xie et al. [5].

The pretrained ResNet18 used in this article was trained several times under different hyperparameter settings (learning rate and batch size) and with a constant number of epochs. In the evaluation of the test dataset, special attention has been paid to the defect detection by considering the TPR, TNR, and F1-score at different learning rates and batch sizes. Based on our results, we observe that using a relatively low batch size and low learning rate, the best results could be obtained. The optimal training parameters depend strongly on the dataset.

The best F1-score result of 93.9% was obtained with a learning rate of $10^{-4}$ and batch size of 16.

ACKNOWLEDGMENTS

The research work of this article was performed at the Polymer Competence Center Leoben GmbH (PCCL, Austria) within the framework of the COMET-program of the Federal Ministry for Transport, Innovation and Technology and the Federal Ministry of Digital and Economic Affairs with contributions by AT&S (Austria Technologie & Systemtechnik AG). The PCCL is funded by the Austrian Government and the State Governments of Styria, Lower Austria, and Upper Austria.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES
