



Explainable Automated Anomaly Recognition in Failure Analysis: is Deep Learning Doing it Correctly?

This is the peer reviewed (post-print) version of the following article:

Original

Explainable Automated Anomaly Recognition in Failure Analysis: is Deep Learning Doing it Correctly? / Arrighi, L., Barbon Junior, S., Pellegrino, F.A., Simonato, M., Zullich, M.. - 1902:(2023), pp. 420-432. (First World Conference on Explainable Artificial Intelligence, xAI 2023 Lisbon, Portugal Luglio 2023) [10.1007/978-3-031-44067-0_22].

Availability:

This version is available at: 11368/3061718 since: 2023-10-26T07:39:00Z

Publisher:

Springer Nature Switzerland

Published

DOI:10.1007/978-3-031-44067-0_22

Terms of use:

Some rights reserved. More details in the PostPrint rights section below.

License:

Copyright Editore

Publisher copyright

--

(Article begins on next page)

Explainable Automated Anomaly Recognition in Failure Analysis: is Deep Learning Doing it Correctly?

Leonardo Arrighi¹[0009-0006-2494-0349], Sylvio Barbon Junior²[0000-0002-4988-0702], Felice Andrea Pellegrino²[0000-0002-4423-1666], Michele Simonato³[0009-0000-1505-5774], and Marco Zullich⁴[0000-0002-9920-9095]

¹ Department of Mathematics and Geosciences, University of Trieste, Trieste, Italy
`leonardo.arrighi@phd.units.it`

² Department of Engineering and Architecture, University of Trieste, Trieste, Italy
`{sylvio.barbonjunior, fapellegrino}@units.it`

³ ASAC srl, Cessalto (TV), Italy
`michele.simonato@asac.it`

⁴ Department of Artificial Intelligence, University of Groningen, Groningen, the Netherlands
`m.zullich@rug.nl`

Abstract EXplainable AI (XAI) techniques can be employed to help identify points of concern in the objects analyzed when using image-based Deep Neural Networks (DNNs). There has been an increasing number of works proposing the usage of DNNs to perform Failure Analysis (FA) in various industrial applications. These DNNs support practitioners by providing an initial screening to speed up the manual FA process. In this work, we offer a proof-of-concept for using a DNN to recognize failures in pictures of Printed Circuit Boards (PCBs), using the boolean information of (non) faultiness as ground truth. To understand if the model correctly identifies faulty connectors within the PCBs, we make use of XAI tools based on Class Activation Mapping (CAM), observing that the output of these techniques seems not to align well with these connectors. We further analyze the *faithfulness* of these techniques with respect to the DNN, observing that often they do not seem to capture relevant features according to the model's decision process. Finally, we *mask out* faulty connectors from the original images, noticing that the DNN predictions do not change significantly, thus showing that the model possibly did not learn to base its predictions on features associated with actual failures. We conclude with a warning that FA using DNNs should be conducted using more complex techniques, such as object detection, and that XAI tools should not be taken as *oracles*, but their correctness should be further analyzed.

Keywords: Convolutional Neural Network · Explainable Artificial Intelligence · Class Activation Mapping · Faithfulness · Printed Circuit Board

1 Introduction

In the last decade, Deep Neural Networks (DNNs) and, more specifically, Convolutional Neural Networks (CNNs) have been widely adopted to solve several classification and regression problems involving images, consistently achieving state-of-art results on several vision benchmark tasks [1, 2] and often surpassing human accuracy. One field where DNNs have been extensively used in recent times is Failure Analysis (FA). FA can be seen as the activity to identify and trace back the root of failures. A failure is defined as «the termination of the ability of an item to perform a required function» [3]. Human-led FA is often applied *in-situ* by human experts, who carefully examine potentially faulty objects in search of possible anomalies and defects [4]. The examination can be carried out visually or with the aid of other tools or methodologies. While these tests are usually accurate, they can be time-consuming and expensive since they have to be executed by expert figures. Thus, there have been many attempts to design semi-autonomous tools to aid human experts in speeding up their evaluations. In many cases, these systems work on raw images of the objects to be inspected: thus, CNN-based techniques are a natural direction for their development.

In the present work, we start by training a CNN to perform FA on Printed Circuit Motherboards (PCBs) of a model specifically created for the project by the company ASAC srl. We will provide additional information on these boards in Section 3.1. Our initial aim is to furnish a professional figure with an automatic diagnosis specifying whether a given PCB is faulty, possibly highlighting areas the CNN considers defective. The training is performed on several images of these PCBs on the premise that the ground truth fed to the CNN is a boolean value indicating whether the motherboard is defective or not. We are able to obtain a validation-set accuracy of 91.61%.

We follow this initial training by conducting an extensive qualitative and quantitative critical analysis of the results produced by the CNN. Specifically, we aim to provide a proof-of-concept using eXplainable Artificial Intelligence (XAI) tools, namely Class Activation Mapping (CAM) [5] and Gradient-weighted CAM (GradCAM) [6] based techniques, to answer the following research questions:

- i. Is the CNN able to identify components of the defective PCBs? We investigate this aspect by comparing the output of the XAI tools with the underlying ground truth (i.e., the defective components). We show that these outputs do not align well with the actual faulty areas of the PCBs, sometimes highlighting parts of the motherboards containing no circuits or even parts of the background.
- ii. Are these XAI tools *faithful* or *correct*? I.e., are they highlighting areas of the images that are effectively relevant to CNN for producing its output? We show that in 94.5% of our test images, *masking out* the portions of the images which seem most relevant according to those XAI tools do not lead to a change in classification, thus revealing low faithfulness. We continue the analysis by showing that even masking out *all* of the faulty components of the PCBs does not lead to a change in classification, highlighting the ineffectiveness of the CNN to capture areas that are connected to faultiness.

Thus, we conclude that, despite showcasing good results accuracy-wise on a validation set, performing FA using a CNN trained with the sole information that a given PCB is faulty/healthy does not yield semantically good results, as the CNN seems unable to correctly identify defective components of the PCB.

In addition, we provide a warning for the community not always to trust the heatmaps produced by CAM and its variants, as they *could* display a low level of *faithfulness* or *correctness* with respect to the model they are applied to, and can, thus, result in inaccurate representations of the inner functioning of a CNN.

Our code and dataset are available on https://github.com/LeonardoArrighi/PCB_Analysis.

2 Related works

2.1 Failure Analysis on PCBs

Failure analysis on PCBs is a critical procedure for determining the underlying cause of PCB malfunctions. PCBs are widely utilized in electronic devices and may fail due to a variety of reasons, such as design flaws, manufacturing defects, environmental stress, and aging. Typically, these steps encompass visual inspection of the faulty PCB to detect any apparent damage or defects, along with non-destructive testing techniques. FA on PCBs is a crucial process that ensures the reliability and performance of electronic devices.

Lewis [7] exhaustively described the principles of FA applicable to polymers. These principles are therefore suited to the identification of failures in PCBs. In particular, identifying corrosion-related defects, such as rust formation, color changes, and other possible evidence, is primarily performed by visual inspection. Kanimozhi and Krishnan [8] proposed an excellent survey summarizing the state-of-art of automatic FA techniques on PCBs. In particular, as in, e.g., [9], the usefulness of studying techniques based on a visual inspection of the PCB was highlighted. For a thorough review of the PCB failure detection approaches proposed in the last three decades, we refer the reader to Ling and Isa [4].

2.2 Explainable Artificial Intelligence

Many XAI techniques have been proposed in order to make ANNs more transparent, explainable, and interpretable [10–12]. Concentrating on CNNs, of particular interest, are methods focused on *visualizing* internal representations or dynamics learned by the networks. Bach et al. [13] underlined the importance of understanding and interpreting classification decisions of automated image classification systems; they proposed using a layer-wise relevance propagation technique to visualize, by means of a heatmap, the *relevant* part of an input in the decision process of a neural network. Their technique makes use of a linear approximation of the class scoring function in the neighborhood of a data point.

Building on the same ideas of visualization-based XAI tools, Zhou et al. [5] proposed a technique called Class Activation Mapping (CAM); the method aims

to visualize predicted class scores on any image fed to CNNs for classification using a *heatmap* that can highlight which parts of the image are most significant for the network itself to perform the task. Inspired by CAM, numerous other techniques have been proposed to generalize the technique to different DNNs or to improve its precision [6, 14–20].

CAM-based techniques have been used to perform localization in CNNs trained on image classification (i.e., without explicit information on the position of an object within an image): Zhou et al. [5] applied CAM to a classification-trained CNN to perform object detection without using bounding boxes for training. Cheng et al. [21], Inbaraj and Jeng [22] developed a similar idea, expanding the so-called *weakly supervised* object detection concept by exploiting CAM-based techniques.

Despite seeing extended usage, XAI tools are difficult to assess quantitatively. Studies that apply these techniques to DNNs are often limited to qualitative considerations on a subset of their datasets, offering mere visual evaluations, e.g., the overlap between semantic features and the outputs of these tools [23, 24]. An extensive line of research exists to develop quantitative or objective ways of evaluating XAI techniques. Nauta et al. [25] summarized many of these works by proposing a unified framework to review these tools. Popular metrics for CAM include the aforementioned correctness or faithfulness or localization, i.e., the ability to localize instances of objects within the images correctly.

An additional point of concern is that these tools are frequently applied as *oracles* without investigating their correctness. Some works have raised concerns with CAM-based techniques, such as low faithfulness [20, 26], or inability to capture multiple instances of objects in an image [27], or still difficulty in precise instance localization [28]. These considerations serve as a base for a critical usage of CAM-based techniques.

3 Materials and Methods

3.1 The dataset

To begin the present work, we constructed a dataset of images of PCBs specifically designed to run lifetime testing. All standard updated technical features are presented and suitable for investigation/testing. For the present research, three features have been investigated. We considered circular welding bases (used to assemble discrete components such as power line capacitors), rectangular welding bases (used to assemble Surface-Mount Device components such as micro-controllers), and parallel electrical tracks (used to transfer signals). Specifically, such electrical tracks are designed with different interspace distances, distributed on single and multi-layers, to cover all configurations used in actual PCBs.

The dataset was constructed starting from a batch of 7 PCBs of the described model. Six of these boards were rendered faulty by subjecting them to an aging process: they were immersed in a saline solution (5% salt) for 96 hours, observing the procedure described in [29]. The remaining one was left untouched. This

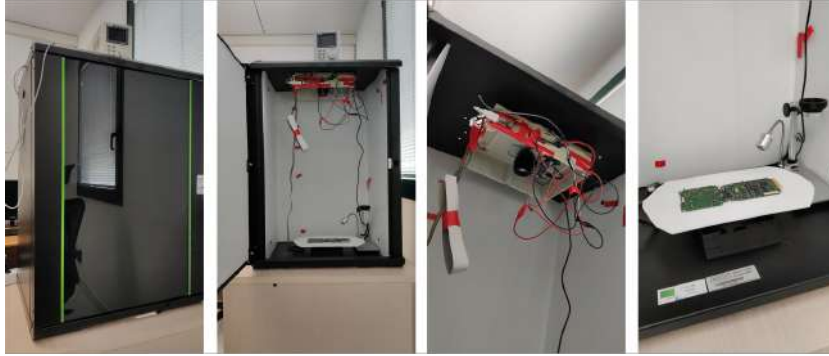


Figure 1: The structure assembled to take pictures for creating the dataset. It is composed of a rack server as a dark room, a Raspberry Pi 4 with which the whole mechanism is controlled, a Raspberry camera with a 1:1 (magnification of the lens) lens, an RGBW LED lamp mounted on the top of the rack server, a LED lamp mounted on the side of the rack server, and a 3D printed rotating platform.

procedure aimed at reproducing faults to the PCBs, which can be attributed to age/corrosion in a realistic usage setting.

In order to artificially increase the number of samples, each PCB image was acquired using a rotating platform where we placed one PCB, a fixed camera, and a set of lights. We could control the orientation and rotation of the board with respect to the camera and toggle light conditions. Some pictures of the structure assembled to take pictures can be seen in the Figure 1. Each board was divided into 4 regions of equal size. Each picture included only one specific region of the PCB. After each shot, the board was rotated by 10° (without contextually rotating the camera), and the next shot of the same region was taken. The whole method was repeated three times: we introduced two sets of different light colors in distinct positions, which were combined in different ways: (i) one white lamp above the board, on the axis of rotation of the platform, and (ii) a natural soft yellow placed on the cabinet wall to form an angle of 30° with the center of the platform, of which we modulated the intensity. This process was aimed at reproducing different light conditions and shot angles. We took the pictures at a resolution of 1024×768 pixels. An example of the images obtained using this protocol can be appreciated in Figure 2. Our goal was to have models, which we trained on this dataset, robust with respect to various factors (such as light conditions and object pose). This would enable the final user to employ it without using a standardized protocol for shooting pictures, even concentrating solely on specific regions of the PCB. We remain available to provide additional information concerning the creation of the dataset for reproducibility purposes.

Once the images were obtained, they were classified as “Defective” (3207 images) or “Non-defective” (2653 images) according to a visual inspection of the components carried out by an expert. We further divided the dataset into

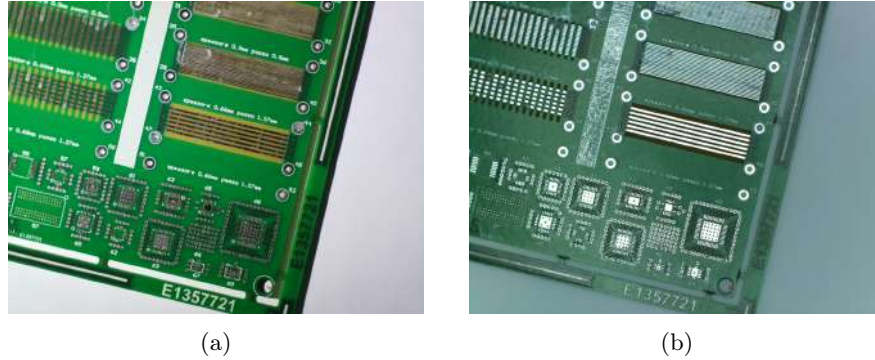


Figure 2: Examples of two images composing our dataset: both pictures depict the same PCB with the same pose with respect to the camera but with different light conditions, which significantly alter its appearance.

a train, validation, and test split: 70% of the images were randomly allocated to the training set (for a total of 4102 pictures), leaving 20% of them in the validation split, and 10% of them in the test set.

3.2 CNN for PCB classification

For performing the PCB classification task, we used a ResNet50d CNN [2, 30], characterized by a sequential stack of residual blocks. Residual blocks are structures containing multiple convolutional layers. The input to a block is *cloned* along two paths, the first being processed sequentially by the convolutional layers and the second being left untouched (*skip connection*), or, in some instances, downsampled by means of other convolutional layers. The data in the skip connection is then summed to the output of the last convolutional layer of the first path. The residual block is designed as in Figure 3. This architecture stacks 12 of these blocks for a total of 50 convolutional layers. Finally, a fully-connected layer for performing the final classification is applied.

3.3 Implementation details

We conducted the experiments on a computing node with an Nvidia Tesla V100 GPU, employing Python version 3.8.12 and the `Pytorch`⁵ library. We fine-tuned the ResNet50d pre-trained model on ImageNet [31]. We recovered the pre-trained parameters released alongside the `timm` library⁶. We made use of the RAdam optimizer [32]. We trained the model for 300 epochs. We set the batch size to 64. We employed dropout [33] with the dropout rate set to 0.1. We used the categorical cross-entropy loss function. Considering that the dataset was built

⁵ <https://pytorch.org/>

⁶ <https://github.com/huggingface/pytorch-image-models>

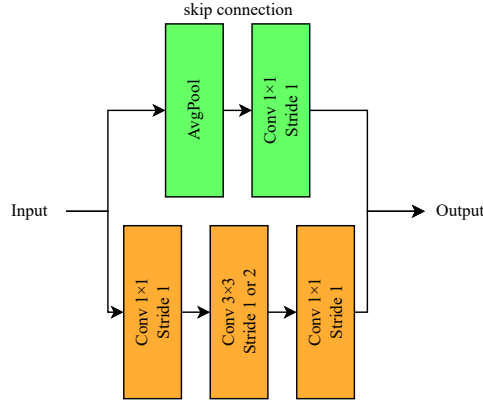


Figure 3: The schematization of the residual block used in the ResNet50d. The input is duplicated along the two paths. The bottom path passes through a sequence of 3 convolutional layers (orange) with different kernel sizes and stride values. The upper path (the skip connection) is left untouched unless a down-sampling of the input is needed: in this case, a sequence of two layers, depicted in green,—an average pooling (“AvgPool”) and a convolutional layer of kernel size 1 and stride 1—are applied.

by encompassing a certain variability in rotation and light conditions, we used a small set of additional data augmentation techniques: horizontal and vertical flip and planar homography.

3.4 XAI techniques used

We used several XAI tools to assess whether the trained CNN correctly identifies features relevant to the FA task on our PCBs. The various techniques adopted are based on CAM [5], an XAI technique specific for CNNs used to display, through an approximate activation map, the parts of an image the CNN has used to classify that specific image in a specific category. According to the definitions provided by Dwivedi et al. [34], CAM, and CAM techniques are *local post hoc* XAI tools. *Post hoc* means they are applied to a model *after* it has been trained without modifying its parameters, while *local* indicates that they produce explanations for single data points which are input to the CNN, rather than giving explanations which are valid globally for the whole model, regardless of its input.

The first relevant technique employed is GradCAM, proposed by Selvaraju et al. [6]. It produces an activation map G_C given by the linear combination of the feature map activations of the last convolutional layer of the network (A_k), weighted by the gradients of the loss function with respect to the neurons of this

layer (w_k^C):

$$G_C = \text{ReLU} \left(\sum_k w_k^C \cdot A_k \right). \quad (1)$$

GradCAM++ [14] is a more advanced version of GradCAM: it calculates the weights w_k using a more elaborate procedure, which includes both the aforementioned activations and gradients of the loss function.

Axiom-based GradCAM (XGradCAM) [18] integrates into the generation of the map a series of axioms such as sensitivity (i.e., that each component of the explanation should be proportional to the impact of removing the corresponding feature from the input on the model’s output) or conservation (i.e., the property that the sum of the components of the explanation should match the magnitude of the model’s output).

LayerCAM [19] generates attention maps for all layers of the CNN, employing GradCAM, and then combines them into a single attention map for the whole image. Thus, LayerCAM provides an interpretation of the model’s decision-making process that considers the contribution of each layer of the CNN.

High-Resolution CAM (HiResCAM) [20] aims at improving GradCAM. The latter calculates weights by averaging gradients (Equation (1)); this leads to an approximation of the heatmap, corresponding to a deficiency in its representation. The attention map in HiResCAM is generated by element-wise multiplying the gradient and the activations (as in Equation (1)) but without computing the average of the gradients.

FullGrad [15], is fundamentally different. It generates heatmaps by computing the gradients of the output of the CNN with respect to both the input image and the bias term in the network. It does not depend on the class used to compute the gradient being propagated; it can produce similar outputs across different classes.

AblationCAM [16] proposes a substantial change to the way the map is generated: the weights of the Equation (1) are calculated using the *slope*, i.e., a function defined as the measure of the effect of ablation of unit k (i.e., an individual activation cell in the feature map A_k of the final convolutional layer of the CNN) on the class activation score y .

Finally, EigenCAM [17] is based on a hypothesis: important spatial features in the input image that the CNN model learns are preserved during the optimization process. Non-relevant or unnecessary features are either smoothed out or regulated, ensuring only relevant spatial features are used to generate the localization map. The activation map of EigenCAM is generated by projecting the output of the last convolutional layer of the CNN onto the first eigenvector of its singular value decomposition.

3.5 Evaluation metrics

We evaluated our model on accuracy (i.e., the fraction of images correctly classified by the model) and the area under the receiver operating characteristics

Table 1: Metrics (Table 1a) and confusion matrix (Table 1b) for the model ResNet50d evaluated on the validation dataset.

(a) Metrics evaluation		(b) Confusion matrix		
Metric	Value (%)	Ground truth	Prediction	
			Defective	Non-defective
Accuracy	91.61	Defective	318	27
AUC-ROC	97.78	Non-defective	25	250

curve (AUC-ROC), a popular metric for binary classification problems. In addition, we provide the confusion matrix to quickly quantify the number of false positives and negatives. All these quantities were calculated on the validation split of the dataset.

4 Results

The performances of the proposed ResNet model are summarized in the Table 1a. The model achieved an accuracy of 91.61%: as seen from the Confusion Matrix (Table 1b), 568 out of 620 test set images are correctly classified. Furthermore, an AUC-ROC of 97.78% reinforces the conclusion that the model behaved effectively.

Among the 318 images of the test set correctly classified by the network as “Defective”, we randomly selected 10 of them to perform the following investigation. We analyzed those pictures with the CAM techniques described in Section 3.4 using the implementations by Gildenblat and contributors [35]. We evaluated the output of the CAM models choosing as target the last convolutional layer of the ResNet50d.

As we can notice in Figure 4, the first aspect is that the CNN is apparently unable to identify specific patterns that refer to the failures: semantically meaningless areas are highlighted by the various techniques. In many cases, these areas are not even located within the boundaries of the PCBs. At this point, we were interested in determining whether these areas represent relevant features for the classification by the CNN. As mentioned in Section 2.2, we removed the topical areas identified by CAM-based techniques. Following a visual inspection, we set a threshold at 0.4 and binarized the outputs of CAM-based methods according to this value, thus creating a binary mask that identifies the topical regions. We then applied these masks to the original images by replacing the pixels of these areas with specific colors/patterns, as shown in Figure 5. After the masking, the new dataset comprised 400 pictures. We re-evaluated these images on the CNN. The result is that 22 of the 400 photos (5.5%) were reclassified as “Non-Defective”, while the remaining ones (94.5%) were still classified as “Defective”. As a last test, to check whether the CNN was learning relevant features for the

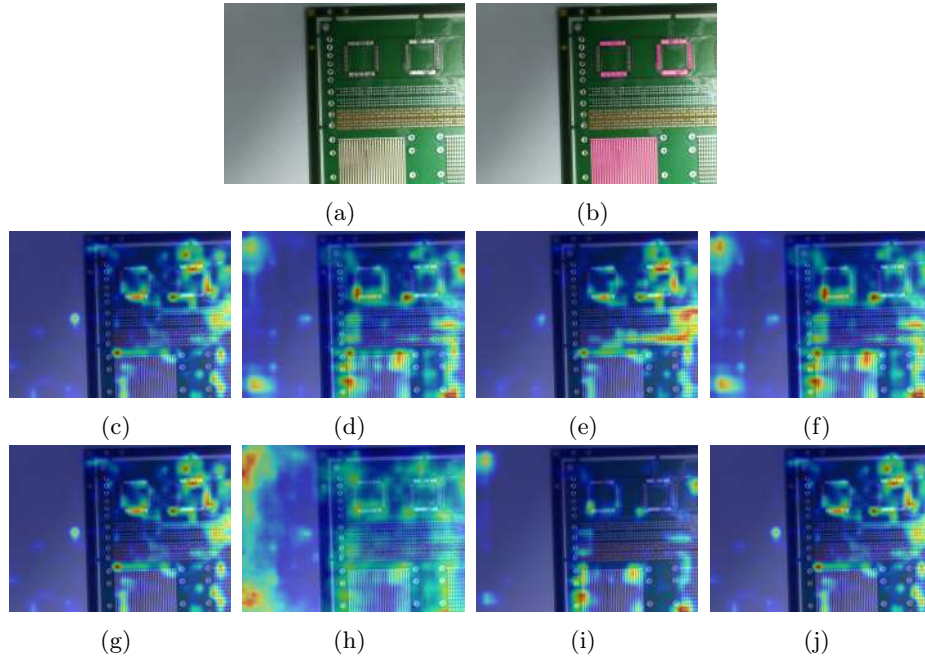


Figure 4: Examples of a single image analyzed through CAM-based techniques: the picture of a region of a PCB (a), classified as “Defective” due to the defective components highlighted in (b), is overlapped with the heatmaps generated by GradCAM (c), GradCAM++ (d), AblationCAM (e), LayerCAM (f), XGradCAM (g), FullGrad (h), EigenCAM (i), and HiResCAM (j).

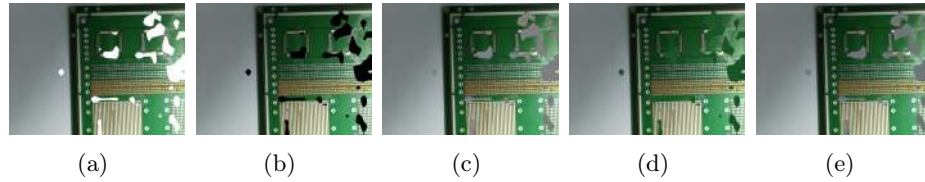


Figure 5: Examples of masks applied to original images by transforming GradCAM output. The patterns adopted are: all white (a), all black (b), random pixels (c), the color of the green surface of the board (d), and the color of the background of the area outside the board (e).

classification, we operated directly on the original 10 images by masking out the faulty components with the same 5 modalities mentioned above. Of the 50 images thus obtained, 45 were evaluated as “Defective” by the model. Therefore, in 90% of the images, the classification did not change. This shows how the CNN, originally thought to perform well on our dataset, is, on a deeper level, unable to learn features connected to true failures.

5 Conclusions

In this paper, we proposed using a Deep Neural Network (DNN) to identify failures on Printed Circuit Motherboards (PCBs) by classifying images of portions of the boards as flawless when there were no evident defects and as defective when there was a visible defect. We showed how we obtained a seemingly accurate and robust binary classifier with a Convolutional Neural Network (CNN), specifically a ResNet50d. We also provided a proof-of-concept using Class Activation Mapping (CAM)-based eXplainable Artificial Intelligence (XAI) techniques to investigate whether the model could identify defective PCB components within pictures of portions of motherboards. We showed that, by and large, the techniques we employed did not correctly highlight the defective areas of the boards. Furthermore, we proceeded to assess the faithfulness of these techniques with respect to the decision process of the CNN. We masked out the areas emphasized by the CAM-based techniques in the test images. We found that these areas did not correspond to significant features used by the CNN for the classification. We also proceeded to mask out the defective PCB components in a selection of test images, showing that this did not significantly change the classification operated by the CNN. Based on the evidence, it can be inferred that the Convolutional Neural Network (CNN) utilized in the analysis may not possess an accurate understanding of the concept of failure. This suggests that the classification of the images may have been influenced by various visual cues and features identified by the model.

Our findings prompt us to posit that Failure Analysis using DNNs should be conducted using more suitable and complex methods like object detection or segmentation to capture the defective components better and that the sole PCB-level information of defect is insufficient for training a well-functioning model. Furthermore, we conclude with a call towards a more responsible usage of local post hoc XAI tools, specifically CAM-based techniques: they should not be regarded as *oracles* since, as we showed in the present work, their outputs might be misleading. We advocate in favor of a critical usage of such tools—appropriate in-depth analyses should always be carried out to confirm their reliability.

References

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. <https://doi.org/10.1145/3065386>.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [3] Marvin Rausand and Knut Øien. The basic concepts of failure analysis. *Reliability Engineering & System Safety*, 53(1):73–83, 1996.
- [4] Qin Ling and Nor Ashidi Mat Isa. Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey. *IEEE Access*, 11:15921–15944, 2023. <https://doi.org/10.1109/ACCESS.2023.3245093>.
- [5] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.
- [6] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [7] Peter Rhys Lewis. Sample examination and analysis. In Peter Rhys Lewis, editor, *Forensic Polymer Engineering (Second Edition)*, pages 33–69. Woodhead Publishing, 2016. ISBN 978-0-08-101055-6. <https://doi.org/10.1016/B978-0-08-101055-6.00002-1>.
- [8] S Kanimozhi and K Gopala Krishnan. A review on automatic bare pcb board testing. *International Journal of Science & Technology ISSN (online)*, 2013.
- [9] E. Salahinejad, R. Eslami-Farsani, and L. Tayebi. Corrosion failure analysis of printed circuit boards exposed to h₂s-containing humid environments. 79: 538–546, 2017. ISSN 1350-6307. <https://doi.org/10.1016/j.engfailanal.2017.05.038>.
- [10] Plamen P. Angelov, Eduardo A. Soares, Richard Jiang, Nicholas I. Arnold, and Peter M. Atkinson. Explainable artificial intelligence: an analytical review. *WIREs Data Mining and Knowledge Discovery*, 11(5):e1424, 2021. ISSN 1942-4795. <https://doi.org/10.1002/widm.1424>.
- [11] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bannetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. 58:82–115, 2020. <https://doi.org/10.1016/j.inffus.2019.12.012>.
- [12] Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. 2023. ISSN 1573-756X. <https://doi.org/10.1007/s10618-022-00867-8>.

- [13] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015. <https://doi.org/10.1371/journal.pone.0130140>.
- [14] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-CAM++: Improved visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision*, pages 839–847, 2018. <https://doi.org/10.1109/WACV.2018.00097>.
- [15] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. *Advances in neural information processing systems*, 32, 2019. <https://doi.org/10.5555/3454287.3454658>.
- [16] Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 983–991, 2020. <https://doi.org/10.1109/WACV45572.2020.9093360>.
- [17] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2020.
- [18] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *arXiv preprint arXiv:2008.02312*, 2020.
- [19] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021. <https://doi.org/10.1109/TIP.2021.3089943>.
- [20] Rachel Lea Draelos and Lawrence Carin. Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks. *arXiv e-prints*, 2020.
- [21] Gong Cheng, Junyu Yang, Decheng Gao, Lei Guo, and Junwei Han. High-quality proposals for weakly supervised object detection. 29:5794–5804, 2020. ISSN 1941-0042. <https://doi.org/10.1109/TIP.2020.2987161>. Conference Name: IEEE Transactions on Image Processing.
- [22] X. Alphonse Inbaraj and Jyh-Horng Jeng. Mask-GradCAM: Object identification and localization of visual presentation for deep convolutional network. In *6th International Conference on Inventive Computation Technologies*, pages 1171–1178, 2021. <https://doi.org/10.1109/ICICT50816.2021.9358569>.
- [23] Jessica Fernandes Lopes, Victor G. Turrissi da Costa, Douglas F. Barbin, Luis Jam Pier Cruz-Tirado, Vincent Baeten, and Sylvio Barbon Junior. Deep computer vision system for cocoa classification. *Multimedia Tools and Applications*, 81(28):41059–41077, 2022. <https://doi.org/10.1007/s11042-022-13097-3>.
- [24] Ho-Taek Joo and Kyung-Joong Kim. Visualization of deep reinforcement learning using grad-cam: how ai plays atari games? In *2019 IEEE Confer-*

- ence on Games (CoG), pages 1–2. IEEE, 2019. <https://doi.org/10.1109/CIG.2019.8847950>.
- [25] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *ACM Comput. Surv.*, feb 2023. ISSN 0360-0300. <https://doi.org/10.1145/3583558>.
 - [26] Jun Li, Daoyu Lin, Yang Wang, Guangluan Xu, and Chibiao Ding. Towards a reliable evaluation of local interpretation methods. 11(6):2732. ISSN 2076-3417. <https://doi.org/10.3390/app11062732>.
 - [27] Prateek Singh and Ambalika Sharma. Interpretation and classification of arrhythmia using deep convolutional network. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022. <https://doi.org/10.1109/TIM.2022.3204316>.
 - [28] Mengying Xiao, Liyuan Zhang, Weili Shi, Jianhua Liu, Wei He, and Zhenggang Jiang. A visualization method based on the grad-cam for medical image segmentation model. In *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pages 242–247, 2021. <https://doi.org/10.1109/EIECS53707.2021.9587953>.
 - [29] M. Mostofizadeh, J. Pippola, T. Marttila, and L. K. Frisk. Effect of thermal aging and salt spray testing on reliability and mechanical strength of sn–58bi lead-free solder. *IEEE transactions on components, packaging and manufacturing technology*, 3:1778–1785, 2013. <https://doi.org/10.1109/TCPMT.2013.2267333>.
 - [30] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 558–567. IEEE. ISBN 978-1-72813-293-8. <https://doi.org/10.1109/CVPR.2019.00065>.
 - [31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. <https://doi.org/10.1109/CVPR.2009.5206848>.
 - [32] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.
 - [33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. URL <https://dl.acm.org/doi/10.5555/2627435.2670313>.
 - [34] Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, and Rajiv Ranjan. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Comput. Surv.*, 55(9), jan 2023. <https://doi.org/10.1145/3561048>.
 - [35] Jacob Gildenblat and contributors. Pytorch library for cam methods. <https://github.com/jacobgil/pytorch-grad-cam>, 2021.