TABLE OF CONTENT

ABSTRACT

Printed Circuit boards (PCBs) are one of the most important stages in making electronic products. In manufacturing an electronic device, since PCBs are the first step, a simple error leads to considerable flaws in the final product. Hence, considering the magnitude and demand of the PCB industry, it is extremely essential to detect and locate the defects in the manufactured PCBs. In this study, we propose a reference comparison solution that uses a template and a test image and combines machine learning and image processing techniques for PCB defect localization and classification. We introduce a solution pipeline that works in three stages: **Subtraction of Images** where the template and test images are subtracted to generate a mask that highlights the regions of defect on the test image, **Contour search** to find the contour of the defects from the generated test image, and **Model Inference** where the contours found above are extracted from the images and fed into our machine learning model which then classifies those defects in the test image. We train the machine learning model, in our case, a Convolutional Neural Network (CNN) that forms the artificially intelligent (AI) part of our solution which can identify the type of defect present given the image of the defect as input. Our experimental results indicate that our model detects the defective regions with an accuracy of 97% which is better than other conventional methods used for this task.

CHAPTER 1 : Introduction and Literature Review

1.1. What is a Printed Circuit Board (PCB)?

PCB or printed circuit boards are the fundamental building blocks of modern electronic devices. It is a module of electronic components that are interconnected in a self-contained structure. It acts as the base of all other electronic components.

Fig. 1 : A close image of a PCB.

A printed circuit board (PCB) gives mechanical support to the circuit, and it also electrically connects circuit components using conductive tracks that are etched from one or more sheet layers of copper separated between sheet layers of an insulating substrate. Circuit components are generally soldered onto the PCB to both electrically connect and mechanically fasten them to it.

Printed circuit boards are used in almost every electronic product. PCBs are usually designed on a computer software where the circuit is first simulated and tested and then eventually the circuit is printed on the board and some other holes are then punched to make space for the connection of components.

A PCB has several advantages over a conventional electronic circuit which consists of wires :

- Miniaturized circuits – PCB made it possible to fabricate complex circuits on a small area.
- Increased life span – Due to the protection of electrical connections on board with the insulated mask coating, the life of PCBs increases significantly.
- Easy Manufacturing - PCB manufacturing is a relatively inexpensive and simpler task than making a wired connection.
- Easy replication - Once the design is ready, multiple boards can be manufactured with the same design on large scale.

- Easy troubleshooting - Inspection, repairing and troubleshooting of faulty components is very easy on a PCB.
- Completely portable - PCBs can be moved from one place to another easily without any damage to the circuits.
- Clean Construction - PCBs allow clean construction of complex electronic circuits.

The PCB manufacturing process may differ slightly depending on the manufacturer, largely in terms of component mounting techniques, testing methods, and so on. PCBs are manufactured in bulk quantities using various automated machines for drilling, plating, punching, and so on. Except for some small variations, the main stages involved in the PCB manufacturing process are the same, as follows :

- Etching the PCB
- Stripping the PCB
- Solder Resist
- PCB Testing
- PCB Assembling

1.2. Problem Description

Printed Circuit boards (PCBs) are one of the most important stages in making electronic products. In manufacturing an electronic device, since PCBs are the first step, a simple error leads to considerable flaws in the final product. The world market for bare PCBs exceeded $60.2 billion in 2014 and is estimated to reach $79 billion by 2024. Hence, detecting all defects in PCBs and finding them is very important.

It is necessary to detect numerous defects on bare PCBs before the etching process since etching usually contributes most destructive defects found on PCBs. The source of defects comes from the printing process which is done before the etching process. Short and open-circuits fall in the fatal defects category. Meanwhile, the other defects such as pinhole, under etch, mouse bites, and missing hole, fall in the potential defects category. Fatal defects are those defects in which the PCB does not serve the objective they are designed for, while the potential defects are those compromising the PCB during the utilization.

Manual visual inspection is one of the most significant time and cost consuming processes in PCB manufacturing. There are multiple defects to detect with an extremely low tolerance for errors and significant expertise required to reliably recognize and handle flawed units. An automated visual printed circuit board inspection approach is, thus, needed to counter difficulties that occurred in manual inspection that can eliminate subjective aspects and then provides fast, quantitative and dimensional assessments.

1.3. Our Solution

Conventional automated optical inspection (AOI) methods for inspecting printed circuit boards can be divided into 3 main streams [1]: reference comparison, non-reference verification, and hybrid approaches. In reference comparison approach, a standard image called a template will be prepared first, and then a PCB that needs to be inspected will be compared with the template to find the unknown defects. In the non-reference verification approach, the method aims to find out if the wiring track, pad, and hole comply with the design without a template image. This approach does not have the limits of the reference method but it still has many difficulties in detecting large defects. In the hybrid approach, the reference method and non-reference method are merged, this approach will have the merits of the two primary methods, meanwhile, it requires high computation capacity.

In our solution, we use a reference comparison approach since it far exceeds the other two approaches in terms of accuracy and efficiency. To solve the problem of PCB defect classification, we propose an approach that combines machine learning and image processing techniques for PCB defect localization and classification. We train a machine learning model, in our case, a Convolutional Neural Network (CNN) that forms the artificially intelligent (AI) part of our solution which can identify the type of defect present given the image of the defect as input. This model is trained using the DeepPCB dataset, which will be explained in later sections of this report.

After the model is trained, we can plug this model into our main pipeline. First, the template (perfect PCB image) and test (defective PCB image) images are taken as input, which is then passed to the image subtraction stage where these images get subtracted to produce a “difference” image which is used to generate a mask that helps to highlight the regions containing the defect. The mask applied to the test image is then passed to the contour extraction stage which extracts the individual images of each defect. These images of the defects are then passed to the machine learning model which then classifies these images into their corresponding defect classes. Finally, the corresponding defects are labelled on the test image and returned to the user. Further details of our solution can be found later in this report. A flowchart of our approach can be seen in fig 2.

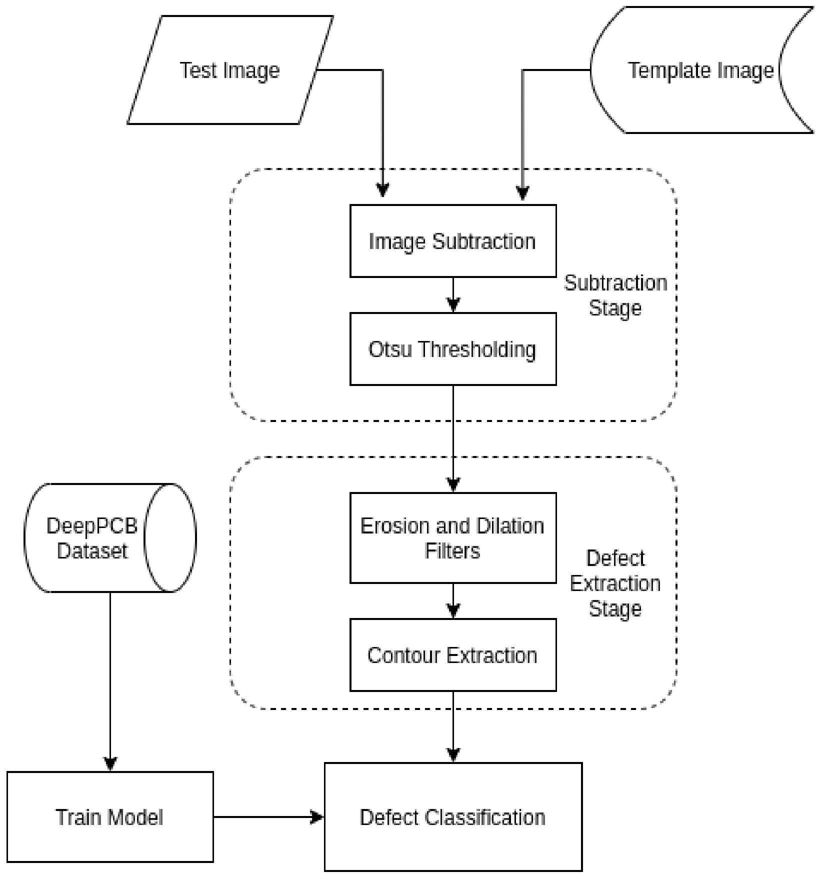

Fig. 2 : The flow chart for PCB inspection. Test image and template will be subtracted and thresholded to locate defects, then these located defects will be send into trained neural network model to get results.

1.4. Literature Review

Various different methods have been proposed in the past for this task. Wen-Yen Wu et al. [1] introduced the development of an automated visual inspection system for PCB which employs a subtraction-elimination method that directly subtracts the template image from the inspected image and then organizes an elimination procedure to locate defects in the PCB. Among these, each defect that is detected is classified by three indices: the type of object detected, the difference in background numbers between the inspected and the template image,and the difference in object numbers. LI Zheng-ming et al. [2] also used digital image-processing technology-based reference methods to classify the defects by getting the number of connected regions, area of defects of the template and inspected image and Euler numbers respectively. Vikas Chaudhary et al. [3] listed 14 kinds of defects that belong to 2 types: positive and negative, and then segmented the image into 3 parts: wiring tracks, soldering pads, and holes. Each defect can be classified by comparing pixels, the number of connected components in the corresponding part. Rudi Heriansyah et al. [4] introduced a new technique that is to classify the defects using the neural network paradigm. Various defective patterns representing corresponding defect types were designed and have employed thousands of defective patterns for training and testing. The result showed the effectiveness of defect classification technology based on neural network.

CHAPTER 2 : Dataset

We use the DeepPCB dataset provided by Tang et. al.[5] for training our models. The dataset contains 1,500 PCB image pairs covering six types of PCB defect. Each pair consists of a 640 x 640 defect-free template image and a defective tested image. We separate 1,300 image pairs as a training set and the remaining 200 image pairs as a test set. Further details about the dataset as described by Tang et. al.[5] are described in the following subsections.

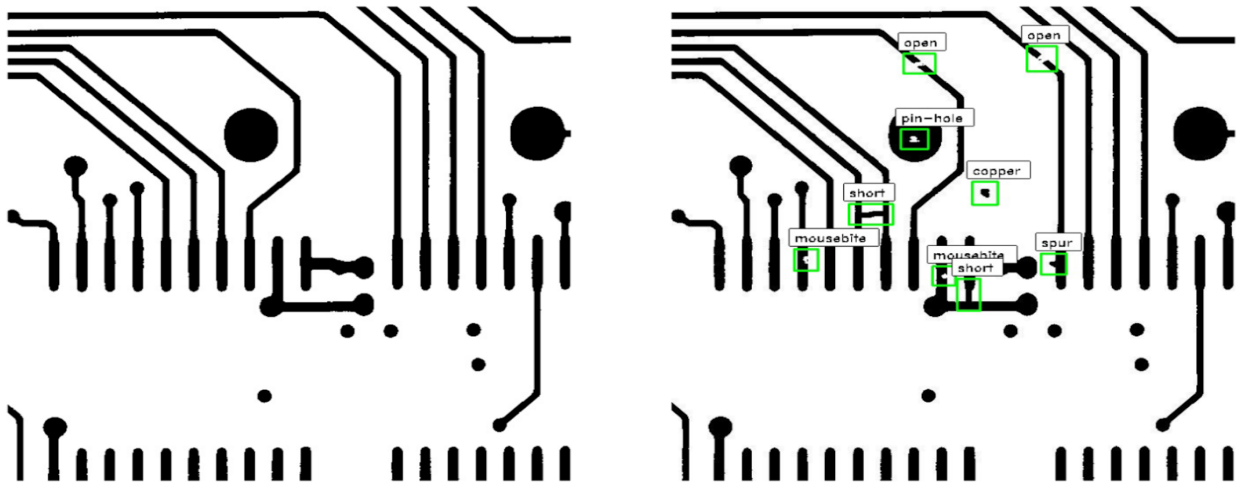

Fig. 3 : This figure shows the pair of (a) a defect-free template image and (b) a defective tested image with annotations of the positions and types of PCB defects in the DeepPCB dataset.

2.1. Image Collection

Having followed the common industrial settings, all the images in this dataset were obtained from a linear scan CCD with a resolution of around 48 pixels per 1 millimetre. Using a sampled image, the defect-free template images are manually checked and cleaned. The original size of the template and tested image is around 16k x 16k pixels. Later they are clipped into sub-images with the size of 640 x 640. For alignment, template matching techniques are used by reducing the translation and rotation offset between the image pairs. A carefully selected threshold is used to employ binarization to avoid illumination disturbance. Apart from different pre-processing methods according to the specific PCB defect detection algorithm, image registration and thresholding are still common techniques for high-accuracy PCB defect localization and classification.

2.2. Image Annotation

An axis-aligned bounding box was used with a class ID for each defect in the tested images. Table 1 shows the six types of PCB defects annotated in the dataset: open, short, mouse bite, spur, pinhole and spurious copper. Since there were only a few defects in the real tested image, the authors had to manually introduce some artificial defects on each tested image according to the PCB defect patterns, which lead to around 3 to 12 defects in each 640 x 640 image. The number of PCB defects is shown in Fig. 4.

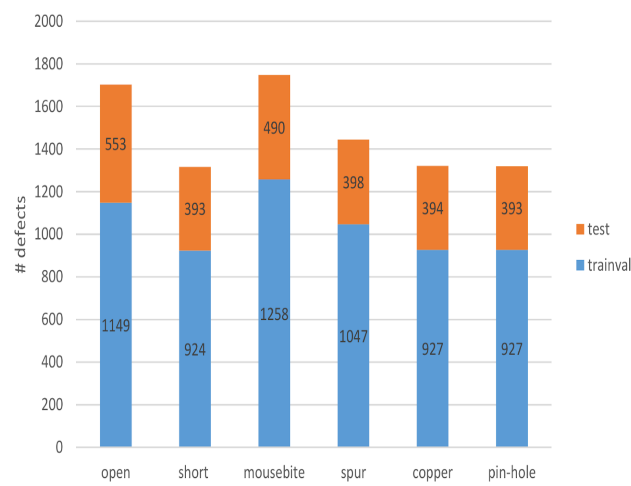

Fig. 4 : Defect number of the 6 categories in DeepPCB train/validation and test set.

No.	Defect
1	Open
2	Short
3	Mousebite
4	Spur
5	Copper
6	Pin-hole

Table 1 : The defects present in the DeepPCB dataset which are also what our solution will learn to detect.

CHAPTER 3 : Methodology

An automated visual printed circuit board (PCB) inspection is an approach used to counter difficulties occurred in human's manual inspection that can eliminate subjective aspects and then provides fast, quantitative, and dimensional assessments. As mentioned before, in this study, we implement a referential approach using test images (i.e. defective PCB images) and template images (i.e. perfect PCB images).

After receiving the input for the template and test images from the user, these images enter into our solution pipeline (Fig. 2) for defect localization and detection which can be broadly divided into three parts :

Subtraction of Images, where the template and test images are subtracted to generate a mask that highlights the regions of defect on the test image,

Contour search, to find the contour of the defects from the generated test image, and

Model Inference, where the contours found above are extracted from the images and fed into our machine learning model (EfficientNet) which then classifies those defects in the test image.

A detailed explanation of the above steps is explained in the following sections.

3.1. Image Subtraction and Thresholding

In the reference comparison approach, a standard image which is called a template will be prepared first, and then a PCB that needs to be inspected will be compared with the template image to find the unknown defects. Although it is straightforward and easy to use, there are various factors that we need to take into consideration like misalignment of the two images, unbalanced illumination, inaccurate registration, vast storage requirements, etc. However, these factors are relatively easier to control owing to the various image registration and acquisition techniques described in the past by papers like [6], [7].

As a first step, after receiving the inputs, the template and test images are subtracted which is essentially used to discover the differences by comparing every pixel value between two images. The difference between two images $f(x,y)$and $h(x,y)$ is expressed as an equation in fig. 5. Consideration of outputs used is just in negative and positive pixel image since zeros values of data do not affect the output of the operation.

$$g(x,y)=f(x,y)-h(x,y)$$

Fig. 5 : Equation representing subtraction operation.

The difference image, thus, obtained after subtraction needs to be converted to a grayscale image before the thresholding operation can be applied to highlight the areas of defect. For the purpose of thresholding, we use global thresholding with the Otsu Binarization technique [8] which enables us to use a dynamic threshold for thresholding. Compared to simple thresholding where an arbitrary threshold is used, Otsu binarization calculates a threshold automatically for bimodal images (In simple words, a bimodal image is an image whose histogram has two peaks due to various noises.), since the difference images obtained earlier are usually bimodal.

3.2. Defect Contour Extraction

After subtraction and thresholding, the red highlighted regions in the test image show the possible areas of defect. Before extracting regions, some erosion and dilation filters are applied to the image to remove any unwanted noise from the image. To extract these regions from the image and feed these into the model, we need to extract the contours of these defects which we implement using OpenCV's find_contour function which stores the coordinates of the boundary of these regions and returns them as lists for each region separately. Using these coordinates we calculate the dimensions of the bounding boxes around these defects. The portions of the image in these bounding boxes is essentially what is extracted from the test image and these smaller images will now be given to the model as input as described in the next section.

3.3. Machine Learning Inference

3.3.1. What is Machine Learning?

Machine learning techniques and solutions have attracted the attention of a variety of people from industry to academia. Machine Learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a subfield of artificial intelligence (AI). Machine learning algorithms usually build a model based on given data, commonly known as "training data", in order to make decisions or predictions without being explicitly programmed to do so. In general, machine learning refers to a set of mathematical tools and techniques that are able to capture patterns, and thus, learn from the data made available to it and make decisions or replicate the pattern learned. One of the most popular machine learning methods is neural networks, which consist of several nodes connected to each other to form a structure like the human brain. The ability of neural networks to generalize most complex functions have made them quite popular for various tasks in many different fields. The branch of machine learning that deals with neural networks is commonly referred to as Deep Learning, which is essentially what we implement in our solution. Deep learning models (or neural networks) employ various optimization techniques which produce a gradient using a convex loss function that back propagates the network to tune the network's weights which, thus, helps the network to learn.

3.3.2 Convolutional Neural Networks

Convolutional neural network (CNN) is a special type of neural network which is used for working with image data. They work on the principle of convolutional filters that are applied across the image to extract spatial features from an image. CNN has a powerful ability to extract features from pictures, and it has been widely used in various computer vision tasks including classification [9],[10], segmentation [11], object detection [12], etc. Lately, a lot of methods based on the convolutional neural network have been adopted, in the field of defect inspection [13]. The results showed drastic improvement in performance compared to the conventional approaches used earlier. As tasks become more and more complicated, the CNNs also become increasingly deep to extract more complicated features that would help in contributing towards the final result. However, there is another problem called gradient diffusion that occurs when the gradient flows back to the beginning if the network is too deep. In this case, most common solution for the above problem is creating a shortcut from early layers to later layers. An architecture based on these shortcuts and other architectural innovations, called the EfficientNet is what we implemented in our solution.

3.3.3 EfficientNet

Released by Google, the EfficientNet [14] architecture has shown to give State-of-the-Art (SOTA) performance on a variety of computer vision-related tasks over the past year. EfficientNet was developed with the key insight that scaling up the width, depth or resolution can improve a network's performance, and a balanced scaling of all three is the key to maximizing improvements. It achieves the state of the art performance with much fewer parameters and FLOPS (floating-point operations per second) than other architectures. Despite these benefits, there have been no studies till now that have applied this architecture for PCB defect classification task. We implement the EfficientNet model using the PyTorch [15] framework and the pytorch-image-models library in Python programming language. The model is trained on the DeepPCB dataset as described in the dataset section. The trained model is then loaded and used at the time of inference where the regions of defect extracted from the test image in the contour extraction step are given as input to the model and the model processes the inputs and assigns probabilities to possible defects mentioned in Table 1. The defect class with the highest probability is thus assigned as a label to that defect and shown to the user in the returned image.

CHAPTER 4 : Observations and Results

4.1. Experimental Setup

The entire project was implemented in the Python programming language and the machine learning models were trained on the NVIDIA Tesla T4 and P100 GPUs provided by Google Colaboratory. The entire code is publicly available at https://github.com/PradyumnaGupta/pcb_defect_analysis.
All the image processing functions used by us including image subtraction, thresholding, filters etc. were used from the OpenCV Python library. For the EfficientNet model, we implemented the original architecture as described in [14] from the pytorch-image-models library. Among the 7 sizes of EfficientNet models available, we used the "b4" version due to the computational constraints. The model was trained using the Adam optimizer [16] with a learning rate of 3e-4 along with a Cross-Entropy loss function [17]. Image augmentations were used to make the model training robust and prevent the model from overfitting. The augmentations used were: normalization, shift-scale-rotate and flip. All the input images were padded to a constant size of 128x128 and the model was trained with a batch size of 32 images for 50 epochs.

4.2. Model Performance

The goal of our solution was to improve the accuracy of the PCB defect localization and classification systems, while also minimizing the time expenditure of the method. As can be seen in Table 2, the model was able to obtain an accuracy of 98.1% on the train set and 97% on the test set. The model took approximately 1 hour to train.

Phase	Metric	Score
Training Phase	Accuracy	0.981
	Cross-Entropy Loss	0.0343
Testing Phase	Accuracy	0.97
	Cross-Entropy Loss	0.11

Table 2 : Results obtained on the training and test sets

CHAPTER 5 : Conclusion and Future Work

PCBs play an important role in producing electronic devices and the quality of the final product depends on its PCB. Therefore, the PCB should be flawless. In this paper, we proposed a defect localization and classification method for PCBs based on complex convolutional neural networks. We trained the network using the DeepPCB dataset and achieved improvement in performance with the EfficientNet architecture. Our results proved the effectiveness of the proposed method. This study presents an approach for defect detection in PCBs, however; the proposed method can be used to detect the defects in other kinds of products such as plastic injection moulding products. Moreover, the subtracting algorithm can be improved to achieve more accurate results in locating the defects.

Future work may focus on continuously increasing the size of the dataset, improving the robustness of the algorithm, reducing the time consumption of the entire detection process while achieving higher efficiency and designing effective non-reference comparison method to avoid using a template.

REFERENCES

1. Wen Yen Wu, Mao Jiun J Wang, and Chih Ming Liu. Automated inspection of printed circuit boards through machine vision.Computers in Industry, 28(2):103–111, 1996.
2. L. I. Zheng-Ming, L. I. Hong, and Jun Sun. Detection of PCB based on digital image processing.Instrument Technique & Sensor, 61(8):87–89,2012
3. Vikas Chaudhary, Ishan R. Dave, and Kishor P. Upla.Automatic visual inspection of printed circuit board for defect detection and classification.Proceedings of the 2017 International Conference on wireless Communications, Signal Processing and Networking, WiSPNET2017, 2018-January:732–737, 2018.
4. Rudi Heriansyah, Syed Abdul Rahman Al-attas, and Muhammad MunimAhmad Zabidi. Neural Network Paradigm for Classification of Defects On PCB.Jurnal Teknologi, 39(1):87–103, 2003.
5. S. Tang, F. He, X. Huang and J. Yang, ”Online PCB Defect Detector OnA New PCB Defect Dataset,” arXiv preprint arXiv:1902.06197, 2019.
6. Huang W, Wei P (2018) A PCB dataset for defects detection and classification. Journal of Latex Class files 14
7. Suhasini, A.. “PCB Defect Detection Using Image Subtraction Algorithm.” (2015).
8. Yousefi, Jamileh. (2015). Image Binarization using Otsu Thresholding Algorithm. 10.13140/RG.2.1.4758.9284.
9. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.Deep residual learning for image recognition. pages 770–778, 06 2016.
10. Gao Huang, Zhuang Liu, Van Der Maaten Laurens, and Kilian QWeinberger.Densely connected convolutional networks.InIEEEConference on Computer Vision and Pattern Recognition, pages 2261–2269, 2017.
11. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation.Arxiv, 79, 11 2014.
12. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, ScottReed, Cheng Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. European Conference on Computer Vision, pages 21–37,2016.

13. Shiyang Zhou, Youping Chen, Dailin Zhang, Jingming Xie, and YunfeiZhou. Classification of surface defects on steel sheet using convolutional neural networks.Materiali in tehnologije, 51:123–131, 02 2017
14. Tan, Mingxing and Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” ArXiv abs/1905.11946 (2019): n. Pag.
15. Paszke, Adam et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library.” NeurIPS (2019).
16. Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” CoRR abs/1412.6980 (2015): n. Pag.
17. Zhang, Zhilu and M. Sabuncu. “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels.” NeurIPS (2018).