Automatic classification of images from PBC Inspection machines – Report 1

The aim of this project is to identify the defective and non-defective patches of PCB acquired with a high resolution camera. Since the PCB can be defective due to different types of problems, we decided to go for a deep learning classification model to classify the defective and non-defective PCB.

Pixellogic Solutions provided a labelled dataset of 6000 samples, of which 3875 were Non Defective and 2125 were Defective. Provided images had dimensions of 600 x 600.

We have created

1) A Training set comprising 2560 samples, of which 1497 are Non Defectiv and 1063 are Defective.
2) A Validation set comprising 462 samples, of which 231 are Non Defective and 231 are Defective
3) A Test set comprising 348 samples, of which 174 are Non Defective and 174 are Defective

There are unused 2630 samples which we have spared for future experiments, of which 1973 samples are Non Defective and 657 are Defective.

Data	**Image Size**	**Number of Defective Samples**	**Number of Non-Defective Samples**	**Total samples**
Training	600 x 600	1063	1467	2560
Validation	600 x 600	231	231	462
Testing	600 x 600	174	174	348

The Test set data (unseen samples which do not contain augmented defected samples) are used to analyze the model performance.

The Model details are as follows :

1) We resized the images to 256 x 256 using Bicubic Interpolation
2) We used the Pytorch implementation of DenseNet201 as the architecture. Instead of training it from scratch, we initialized the parameters with the weights that the model converged to when trained on a popular large dataset called Imagenet. These weights are made available by the Pytorch team publicly. Theoretically, the earlier layers in a neural network model should learn more general representations that should be relevant across datasets. So we freeze the first two “dense “ blocks of the architecture to contain Imagenet pretrained weights. DenseNet201 contains four “dense” blocks, followed by a classifier

layer. Thus the last two “dense” blocks and the classifier layer are fine-tuned on our dataset.

3) We also used an augmentation technique called Cutmix, which randomly replaces patches of a given image with the same patch coordinates of another image. This is done during training. Below, we report the results from using the above model. The values are averaged over 3 sets of experiments. The Defective class is taken to be the positive class

Results

Test Data:

Defected: : 174
Non-Defective : 174

Accuracy (%)	**89.3**
Precision* (%)	**94.9**
Recall (%)	**82.4**
F1 Score	**88.2**

We also provide the results from experimenting with the previous model on this dataset.

Accuracy (%)	**79.4**
Precision (%)	**85.7**
Recall (%)	**71**
F1 Score	**77.6**