|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 balanced out both classes by randomly not considering 1750 Non Defective Samples. That left us with 2125 samples from each class, and a total of 4250 samples, which were further distributed in the following way.  We have created  1) A Training set comprising 3060 samples, of which 1530 are Non Defective and 1530 are Defective.  2) A Validation set comprising 340 samples, of which 170 are Non Defective and 170 are Defective  3) A Test set comprising 850 samples, of which 425 are Non Defective and 425 are Defective   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Data** | **Image Size** | **Number of Defective Samples** | **Number of Non-Defective Samples** | **Total samples** | | Training | 600 x 600 | 1530 | 1530 | 3060 | | Validation | 600 x 600 | 170 | 170 | 340 | | Testing | 600 x 600 | 425 | 425 | 850 |   The Test set data 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. (Cutmix threshold = 0.6)  4) We used the Negative Log Loss function as the objective Function  5) We started training with an initial learning rate of 0.1 and decreased by a factor of 0.1 after every 75 epochs.  6) We trained for 400 epochs although we’d like to point that the same result is obtainable with very few epochs.  We report the results of the three folds.  **Results**  **Test Data:**  Defected: : 425  Non-Defective : 425     |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | Accuracy | Defective Accuracy (Recall) | Non Defective Accuracy  (Recall) | Defective Precision | Non Defective Precision | Defective F1 Score | Non Defective F1 Score | | Fold1 | 86.8% | 88.7% | 84.9% | 85.5% | 88.3% | 87.1% | 86.6% | | Fold2 | 86.1% | 87.5% | 84.7% | 85.1% | 87.2% | 86.3% | 85.9% | | Fold3 | 86.1% | 87.1% | 85.2% | 85.5% | 86.8% | 86.2% | 86% |   We also provide the results from experimenting with the previous model on this dataset.   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | Accuracy | Defective Accuracy (Recall) | Non Defective Accuracy  (Recall) | Defective Precision | Non Defective Precision | Defective F1 Score | Non Defective F1 Score | | Fold1 | 82.2% | 81.4% | 83.05% | 82.7% | 81.7% | 82.08% | 82.38% | | Fold2 | 78.3% | 63.05% | 93.6% | 90.8% | 71.7% | 74.4% | 81.2% | | Fold3 | 79.5% | 80.2% | 78.8% | 79.1% | 79.95% | 79.6% | 79.38% | |