COMPUTER VISION

PCB DEFECT CLASSIFICATION
ASSIGNMENT - 1

SUBMITTED BY Iwrin Show (16CO155)
Chandrika T G (16CO214)
S Chethana Vaisali(16CO255)

Introduction

This work has been carried out as assignment 1 of the course Computer Vision in our 8th semester. Printed circuit board(PCB) is the basic carrier in all electronic devices. The quality of PCB directly impacts the performance of electronic devices. Automated detection of defects of PCB is a problem which many researches are interested in for quite a long time. Three datasets were given to us as a part of the assignment. The problem statement of the assignment is **to classify the PCB image as *defective* or *non-defective***. The rest of the document has sections each about data pre-processing, architecture, results and challenges/problems faced.

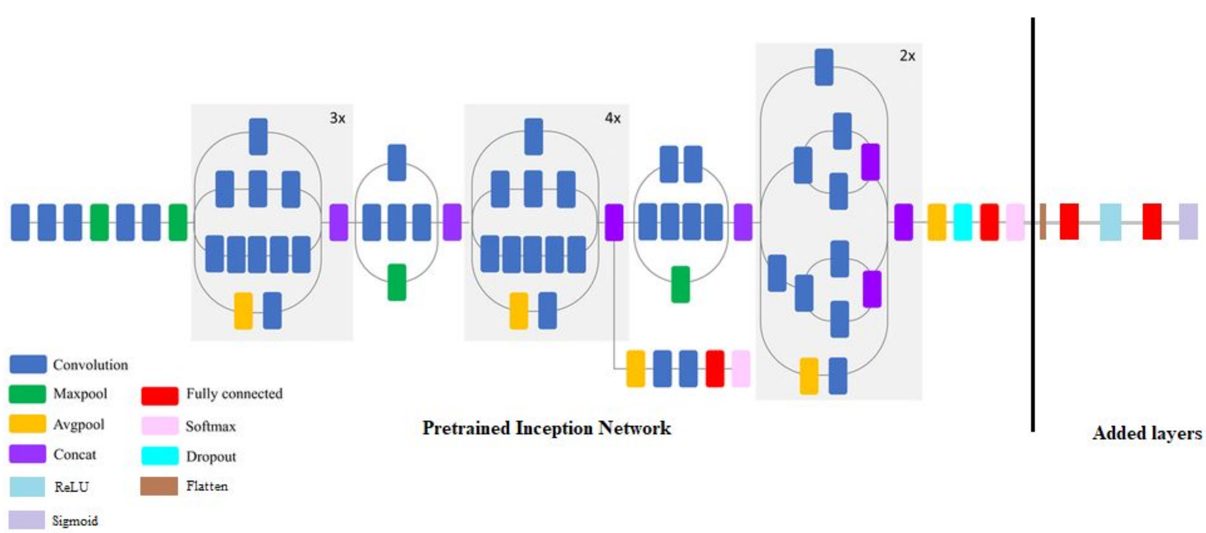
Data pre-processing

Three datasets were given as a part of the problem statement. Among them, **raw dataset** is used for training our model. Following procedure was followed for data pre processing:

- First the data is split into defective and non defective data.
- Test, train and validation sets are split for both defective and non defective data separately in order to ensure that there is no overlapping among all the three sets even in terms of transformations and also to ensure balance between the two classes in all the sets.

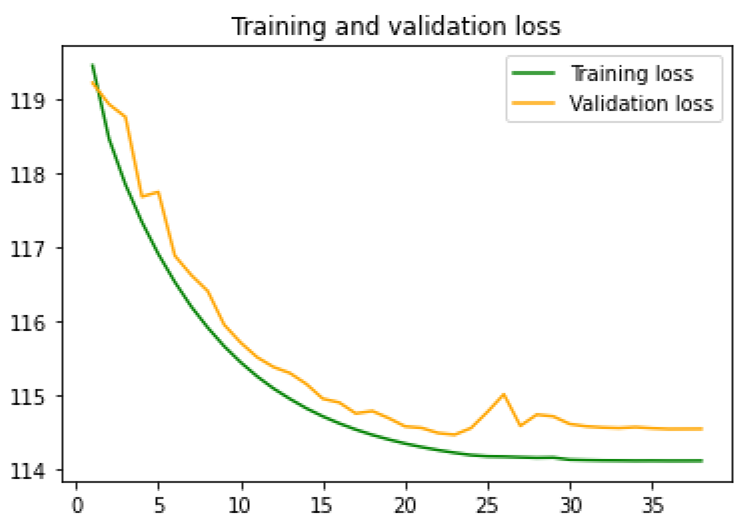
- Since there is high data imbalance between the defective and non defective images where defective samples are less, we *oversampled the train and validation splits of only the defective data* to improve our results.
- Following actions are performed for oversampling the defective data:
    - Horizontal and vertical flipping.
    - +90° and -90° rotation.
    - Noise is added to the images.
- Raw data before oversampling had 149 defective images and after oversampling it is 708.
- Train, test and validation splits of defective data are concatenated with the train, test and validation splits respectively to form the final train, test and validation splits.
- After data pre-processing, following sets are obtained:
    - Train set has **2176 samples** each of shape 300 X 300 X 3.
    - Validation set has **548 samples** each of shape 300 X 300 X 3.
    - Test set has **534 samples** each of shape 300 X 300 X 3.
- Train and validation data are divided by 255 for the purpose of normalisation.

Architecture



As the dataset we are provided with is very small, we used transfer learning for this assignment. We used **the Inception pre-trained model** that is available in Keras. The architecture is as shown in the figure above. **Four layers** are added to the last conv layer of Inception after flattening the input - *Fully Connected layer, ReLU, Fully Connected layer and Sigmoid activation function*. All the layers of the inception base are non-trainable except Batch Normalization layers. As the mean and variance of the current dataset are different from Imagenet dataset, *the Batch Normalization layers are made trainable*. In order to prevent overfitting, **L2 regularization** is used in all the layers with λ = 0.1. **Stochastic Gradient Descent(SGD)** with learning **rate 0.1** and **Nesterov momentum 0.9** is used as the optimizer. Early stopping, ReduceLROnPlateau and ModelCheckPoint are used as callbacks.

The loss graph is shown in the figure below.

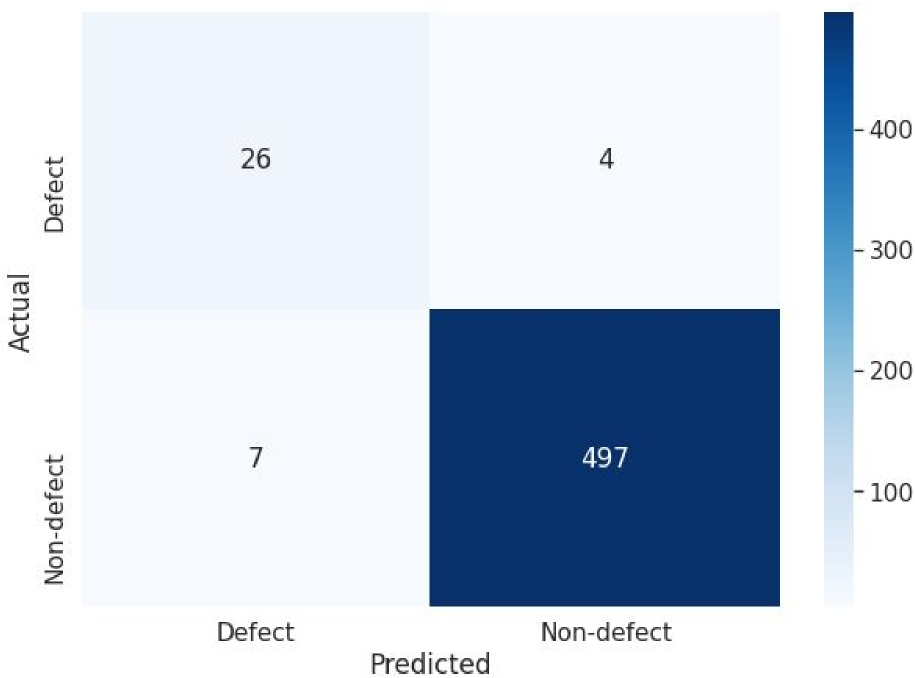


As we see, the model starts overfitting beyond the 23rd epoch. The weights are stored which are used for evaluation.

Results

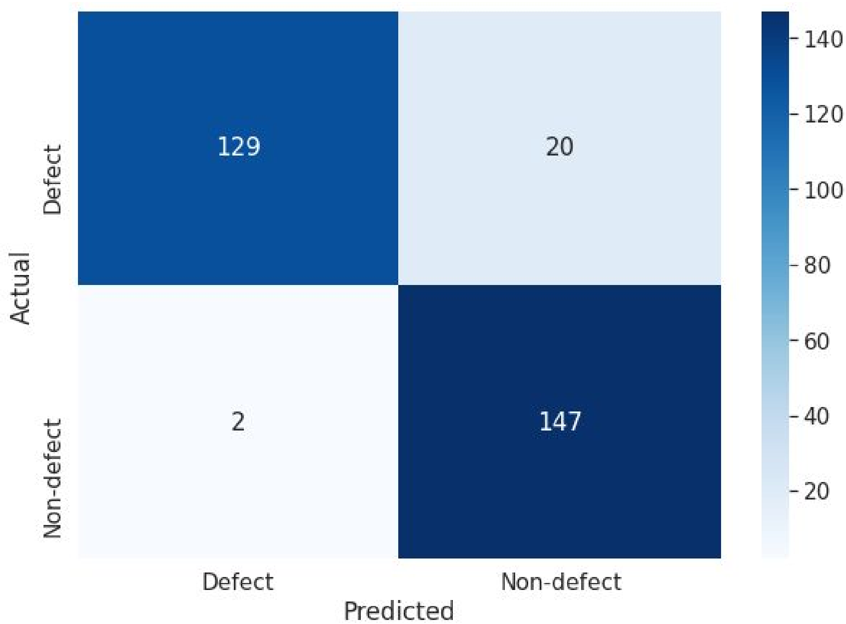
Following is the result table and heat map when the model is evaluated on the test split of the raw dataset.

Test Data	Precision	Recall	Accuracy	F1 score
Defective	0.78788	0.86667	0.86666	0.82540
Non-defective	0.99202	0.98611	0.98611	0.98905
Total accuracy			0.97940	



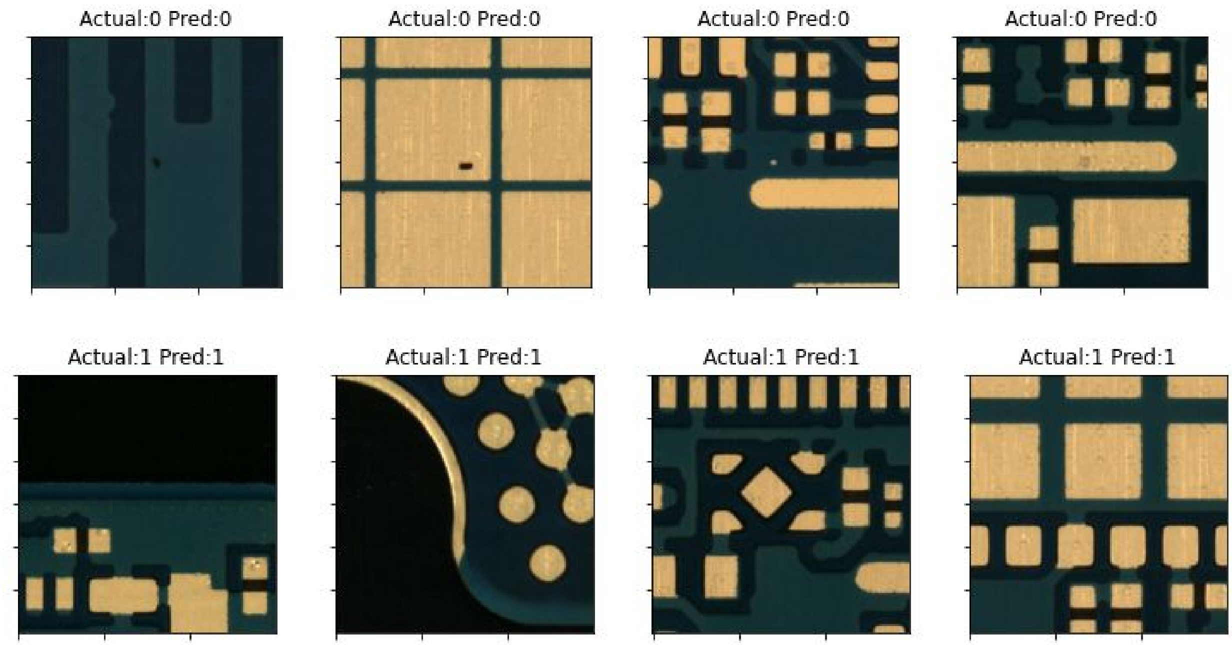
Following is the result table and heat map when the model is evaluated on the entire balanced data.

Balanced Data	Precision	Recall	Accuracy	F1 score
Defective	0.98473	0.86577	0.86577	0.92143
Non-defective	0.88024	0.98658	0.98657	0.93038
Total accuracy			0.92617	

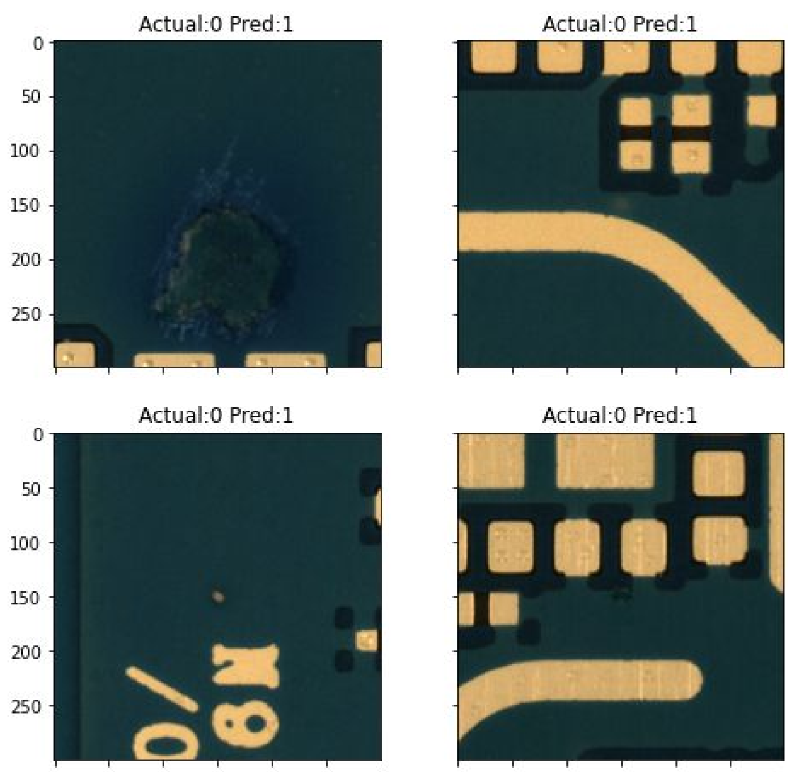


Data	Defective Accuracy	Non-defective accuracy
Raw dataset	0.86577	0.99206
Balanced dataset	0.86577	0.98657

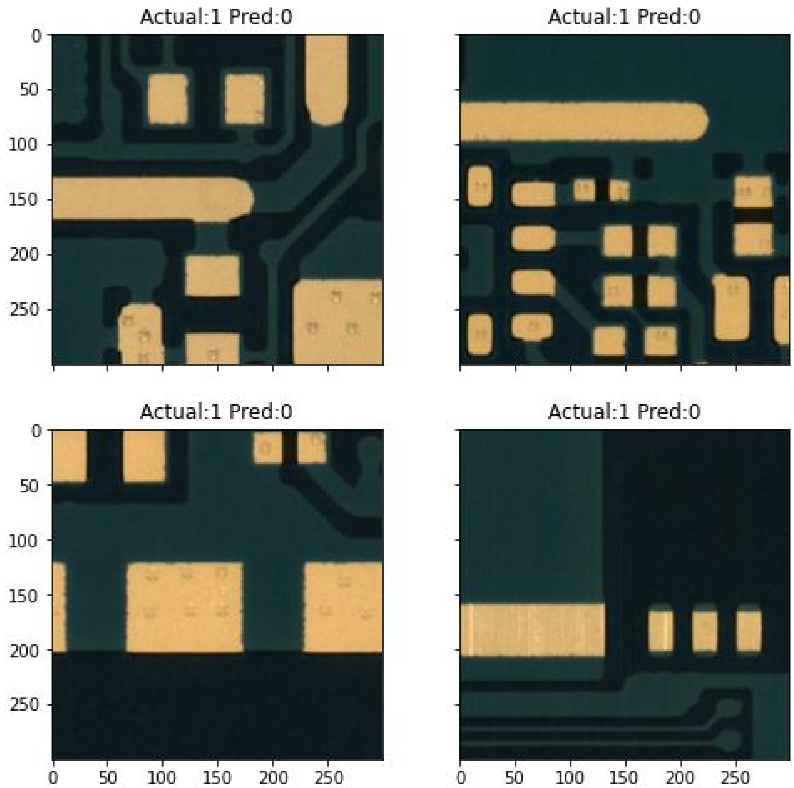
Let us look at some of the visualisations of predictions of the test set.



The above images are some of the images which are classified correctly by the model.



These are some of the images which are defective but are classified as non-defective by the model.



These are some of the images which are actually non defective but are predicted as defective by the model.

Challenges

- The dataset is highly imbalanced. The data samples of the minority class are very less in number. Although the minority class is oversampled, the augmented images are just transformations of the existing images. As such, the model may not perform well on unseen data.