Abstract
The goal is to use Deep Learning in tandem with other techniques to obtain a model that would classify images of printed circuit boards as defective or non defective with a very high accuracy. The literature contains numerous endeavours that leverage the latent space in numerous ways with the end result that they perform really well on benchmark datasets. However, the datasets that the results are reported on are usually, more or less, balanced, in the sense that the counts of class-wise samples are not disproportionately different. So, it seems at the moment, that the problem is not so much about tweaking neural network architectures that would learn rich representations as it is about dealing with the problem of class imbalance.

Introduction

The goal is to build a binary classifier that would map given images of printed circuit boards to one of two classes. At the surface, this doesn't seem like a very novel or challenging task. Machine Learning, and especially Deep Learning, to a very large extent, has been about learning abstract features that would discriminate one class from the other and great efforts have been made in this direction, in devising neural network models. However, authors of these models, more often, report the performance of their models on balanced datasets, that is , datasets where the counts of class-wise samples are either same or differ only slightly.

Real world problems, more often than not (and as is the case with the current task) violate that condition and are imbalanced. To train a model that would tell a defective sample from a non defective one, we need to have samples that are known to be defective (in the supervised learning setting). But the machines that manufacture these boards are highly precise and one should expect the defective samples to be far lesser than the non defective ones. In the dataset used so far, the latter's count is roughly twice that of the former.

This work focuses on techniques that can be used to complement Deep Learning architectures to build models better suited for classification on imbalanced datasets.

Roadmap

The dataset used for the experiments so far, contains 2550 images, each of which belongs to one of two classes, Defective and Non-Defective. The count has been altered to suit the needs of the experiments. Details regarding that are mentioned alongside descriptions of the corresponding experiments. The test set has been kept the same for all the experiments, that is, 174 samples from each class.

Each image has three channels with each channel being a 600X600 matrix. Thus each image has the dimensions 3 X 600 X 600. However the computational resources I have access to are

1) A system with 60GB CPU memory
2) A system with 16GB GPU memory

The above constraints make it impossible to train the images with their original dimensions. Hence, the images had to be resized. At the moment, by downsizing to 3 X 256 X 256, I am able to train the images on GPU with the batch size being 34. Each step in an epoch takes about 570 milliseconds. (Some time was taken up by the effort to figure the downsizing dimensions.)

The nature of the field is perhaps such that any devised technique for a particular problem can be shown to be better than other techniques only by empirically comparing the results of each technique, unlike some other fields where techniques can be theoretically proven to be better. Hence, before working on new devised techniques, it is important to have a record of some baseline models and how they perform on a given task. The work so far has been, and will be, for a few more days, about such baseline models.

Experiments

Since the focus is more on techniques that can be used alongside neural network architectures, for the experiments that I did so far, I chose to stick to one architecture to get an idea of how different techniques work in comparison to one another. The model chosen is the Keras implementation of DenseNet201 initialized with imagenet-trained weights.

Image resizing is done using ‘ Nearest Neighbour Interpolation’ .(Bilinear and Bicubic Interpolation to be tried in future). Another approach for downsizing would be to use an autoencoder where a neural network learns lower dimensional representations.

The dataset is divided into 2 parts with each part comprising samples for

A) Training + Validation [1]
B) Test

1)
Training is done on an imbalanced dataset while keeping the validation set and test set balanced.
Training set has 897 Non Defective and 463 Defective samples.
Validation set has 231 samples from each class.
Test set has 174 samples from each class.
3 Fold Cross Validation is done.
Training is done for 100 epochs with a fixed learning rate of 10^{-3} and using Adam Optimizer.

	Accuracy	Defective Precision	Defective Recall	Non Defective Precision	Non Defective Recall
Fold1 Model	73.27%	72.6%	74.71%	73.96%	71.83%
Fold2 Model	75.28%	85.48%	60.9%	69.64%	89.65%
Fold3 Model	55.17%	62.85%	25.28%	53.23%	85%

2)
Training, Validation and Test sets are balanced.
Training Set is balanced by reducing the count of the majority class to equal that of the minority class.
Training set has 463 samples from each class.
Validation set has 231 samples from each class.
Test set has 174 samples from each class.
3 Fold Cross Validation is done
Training is done for 100 epochs with a fixed learning rate of 10^{-3} and using Adam Optimizer.

	Accuracy	Defective Precision	Defective Recall	Non Defective Precision	Non Defective Recall
Fold1 Model	50.5%	50.35%	80.4%	51.42%	20.68%
Fold2 Model	66.09%	61.76%	84.48%	75.75%	47.70%
Fold3 Model	58.6%	55.35%	89.08%	72.05%	28.16%

3)
Training, Validation and Test sets are balanced.
Training Set is balanced by augmenting the majority class with vertically flipped transformations of already present images to equal that of the minority class.
Training set has 897 samples from each class.
Validation set has 231 samples from each class.
Test set has 174 samples from each class.
3 Fold Cross Validation is done.
Training is done for 100 epochs with a fixed learning rate of 10^{-3} and using Adam Optimizer

	Accuracy	Defective Precision	Defective Recall	Non Defective Precision	Non Defective Recall
Fold1 Model	66.9%	98.3%	34.48%	60.2%	99.4%
Fold2 Model	70.4%	88.1%	47.1%	63.92%	93.67%
Fold3 Model	60.3%	62.5%	51.7%	58.82%	68.96%

4)
Training, Validation and Test sets are balanced.
Training Set is balanced by augmenting the majority class with mirror image transformations of already present images to equal that of the minority class.
Training set has 897 samples from each class. Validation set has 231 samples from each class. Test set has 174 samples from each class.
3 Fold Cross Validation is done.
Training is done for 100 epochs with a fixed learning rate of 10^{-3} and using Adam Optimizer.

	Accuracy	Defective Precision	Defective Recall	Non Defective Precision	Non Defective Recall
Fold1 Model	68.6%	68.78%	68.39%	68.57%	68.96%
Fold2 Model	66.09%	93.75%	34.48%	59.85%	97.7%
Fold3 Model	74.13%	71.85%	79.31%	76.92%	68.96%

Future Work Proposal

1) The same experiments can be repeated by changing the resizing algorithm. Other algorithms include Bilinear and Bicubic Interpolation

2) Resizing can also be done using an Autoencoder.

3) Also, the experiments need to be repeated using both horizontal-flip and vertical-flip transformation augmentation together.

4) The experiments reported here incorporate offline augmentation. Online augmentation needs to be tried.

5) Augmentation techniques like Cutout and Cutmix will be tried.

6) Once some baseline models are trained, more effort will go into hyperparameter tuning

7) More samples are to be incorporated into the dataset.