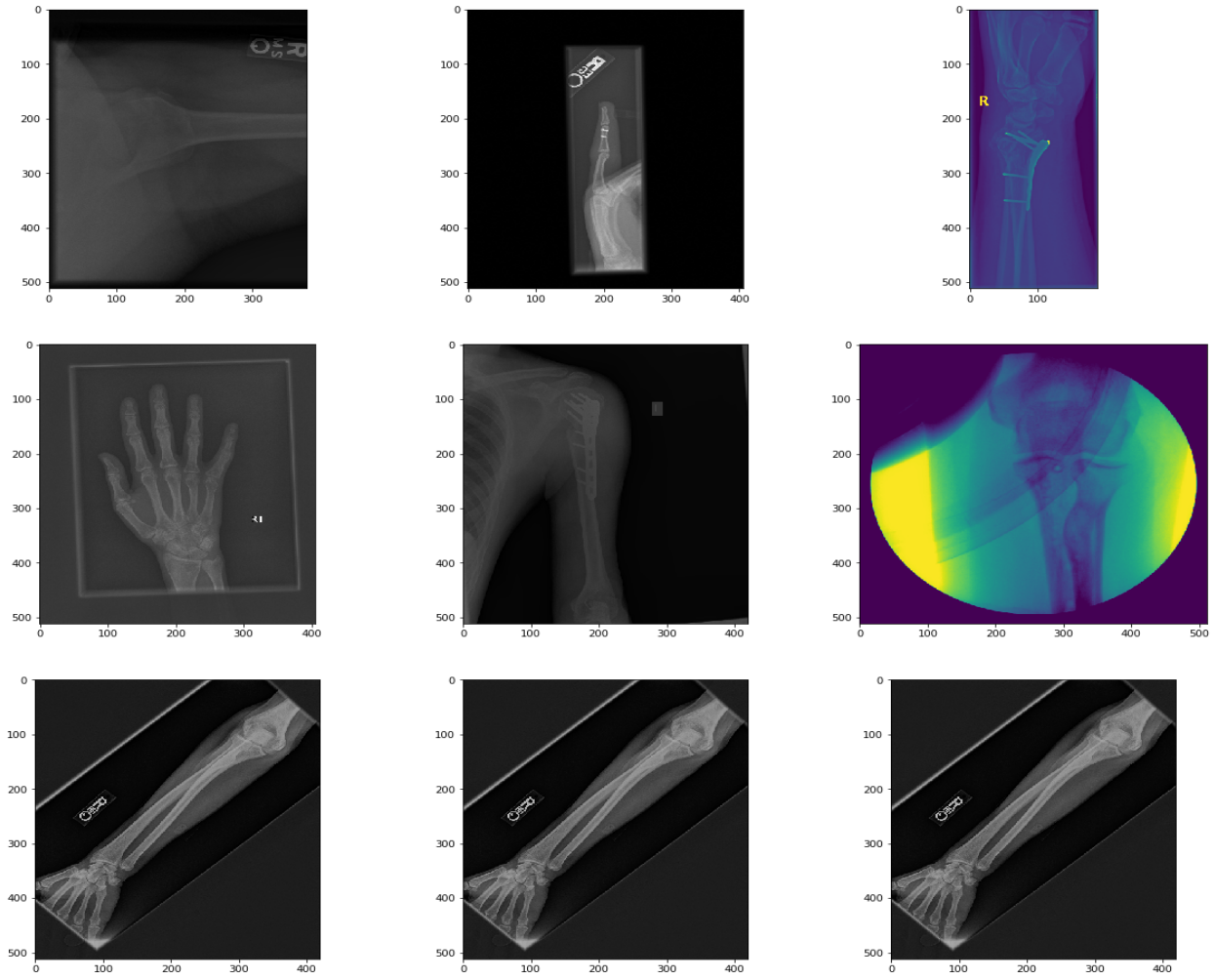
Alexandria University : Computer Vision Fall 2018

Final Project Report

Bone Abnormality Classification in Keras



Zyad Shokry Abozaid **3517**

Kareem Ahmed Abdelsalam **3356**

Khaled Mohamed Khaled **3662**

Introduction

Project Overview

The MURA abnormality detection task is a binary classification task, where the input is an upper extremity radiograph study — with each study containing one or more views (images) — and the expected output is a binary label $y \in \{0, 1\}$ indicating whether the study is normal or abnormal, respectively.

Available X-ray images are of Elbows, fingers, shoulders, forearms, hands, wrists, humerus.

Problem Statement

Implementing a deep convolutional neural networks model that takes a study as input and output the probability of abnormality.

A study consists of one or more images (views) which are feeded one by one into the model to output the probability of abnormality of each one, then the overall probability of a study is the arithmetic mean of all its views.

The model makes the binary prediction of abnormal is the probability of abnormality is greater than 0.5 .

Evaluation Metrics

The primary evaluation metric is a **confusion matrix** with an aim to maximize the True Positives (Abnormals detected as Abnormals) and True Negatives (Normal studies detected as normal) and minimize False positives and Negatives with a priority of optimizing True positives and minimize False positives.

This is due to the nature of the problem (a medical application) where our priority is to detect all Abnormalities and not label an Abnormal study as normal.

<u>One number to judge:</u>

If we wanted only one number to judgethe model : Precision and recall seem to be relevant to measure robustness of the model.

Precision Is the number of true positives divided by the number of possible samples. i.e. (# of Abnormal scans predicted as normal) / # of Abnormal scans in the evaluation dataset Precision measures how good the model is at detecting Abnormalities. Also should calculate precision for Normal scans but for Abnormal ones is more important.

Recall / Sensitivity Is the number of true positives divided by the samples. i.e. (# of Abnormal scans predicted as normal) / # of samples.

Recall measures the randomness in the model, and its best

But:

Although **Accuracy** is not perfectly suitable for our problem we are using it to judge how much our model is performing better than a naive model predicting all as Normal or Abnormal.

The Accuracy is simply the sum of true positives and true negatives divided by the total number of samples. i.e. (Abnormal scans that were predicted Abnormal + Normal scans that were predicted Normal)/ # of samples.

So to summarize : our evaluation will be based on both Accuracy and Confusion Matrices.

Analysis

Data Exploration

Structure of Data:

Dataset is unzipped into 2 folders ("train" and "valid" folders) and 4 csv files

- train folder contains training set images
- valid folder contains validation set images
- train_image_paths.csv contains paths to all images in the training set
- train_labeled_studies.csv contains paths to all studies in the training set, and the second column has the label of the study
- labels are 0 for normal and 1 for abnormal

- valid_image_paths.csv and valid_labeled_studies.csv are like training ones
- A study directory contains one or more png images of views of a certain limp, named imagex.png where x is index of image
- images are organized in folders as SET_FOLDER -> LIMP FOLDER -> PATIENT FOLDER -> STUDY FOLDER -> images (views)

Training Data

Limp	Normal Studies	Abnormal Studies	Study Ratio	Normal Images	Abnormal Images	Images Ratio
Shoulder	1364	1457	0.51648	4211	4168	**0.50256**
Wrist	2134	1326	0.61676	5769	3987	**0.59132**
Finger	1280	655	0.66149	3138	1968	**0.61457**
Elbow	1094	660	0.62371	2925	2006	**0.5931**
Humerus	321	271	0.54222	673	599	**0.52908**
Hand	1497	521	0.74182	4059	1484	**0.73227**
Forearm	590	287	0.6727	1164	661	**0.6378**
Total	8280	5177	0.61529	21939	14873	**0.59597**

Validation Data

Limp	Normal Studies	Abnormal Studies	Study Ratio	Normal Images	Abnormal Images	Images Ratio
Shoulder	99	95	0.5103	285	287	**0.5062**
Wrist	140	97	0.5907	364	295	**0.55235**
Finger	92	83	0.5257	214	247	**0.53579**
Elbow	92	66	0.58227	235	230	**0.50537**
Humerus	68	67	0.5037	148	140	**0.5138**

Hand	101	66	0.6047	271	189	**0.5891**
Forearm	69	64	0.5187	150	151	**0.50166**
Total	661	583	0.55129	1667	1530	**0.5214**

Benchmark

Since the data is not symmetric as seen from the above table, number of Normal samples is usually higher than Abnormal ones, the random chance benchmark of 50% cannot work here.

So we calculated ratios of numbers majority class samples to minority class samples for each limpin training and validation data, where a sample can be an image or a study containing a number of images for the same limp.

The produced ratios indicate the accuracies that we should indeed pass by some acceptable margin in order to have an acceptable model.

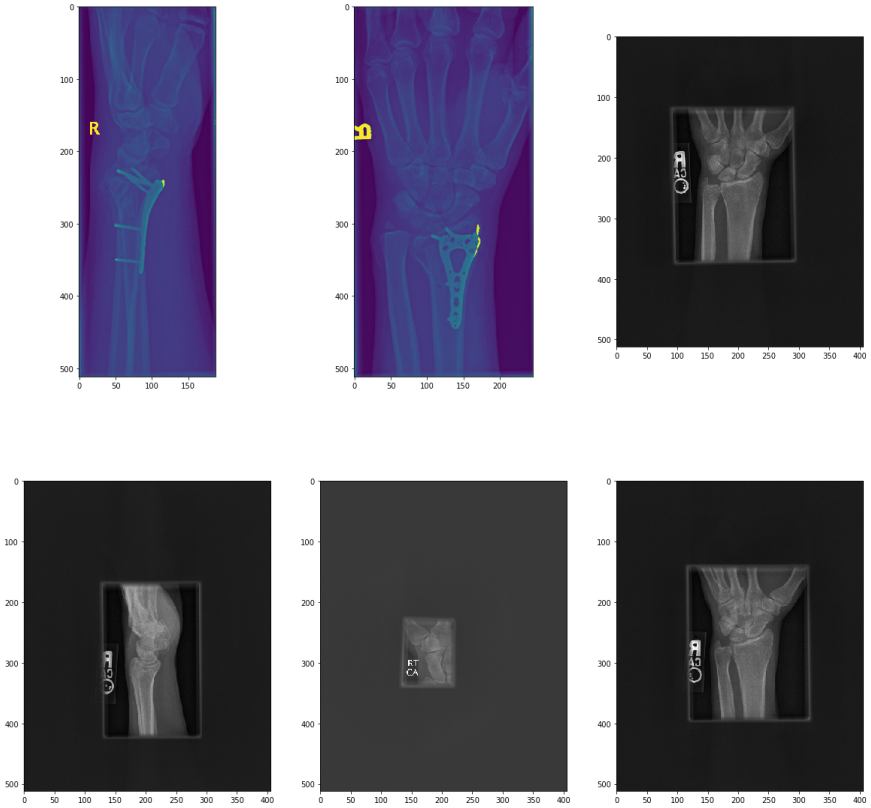
So if we are training on all limbs and evaluating on all of them we should be getting higher accuracies than 60% for training set and 53% for validation set.

Exploratory Visualization

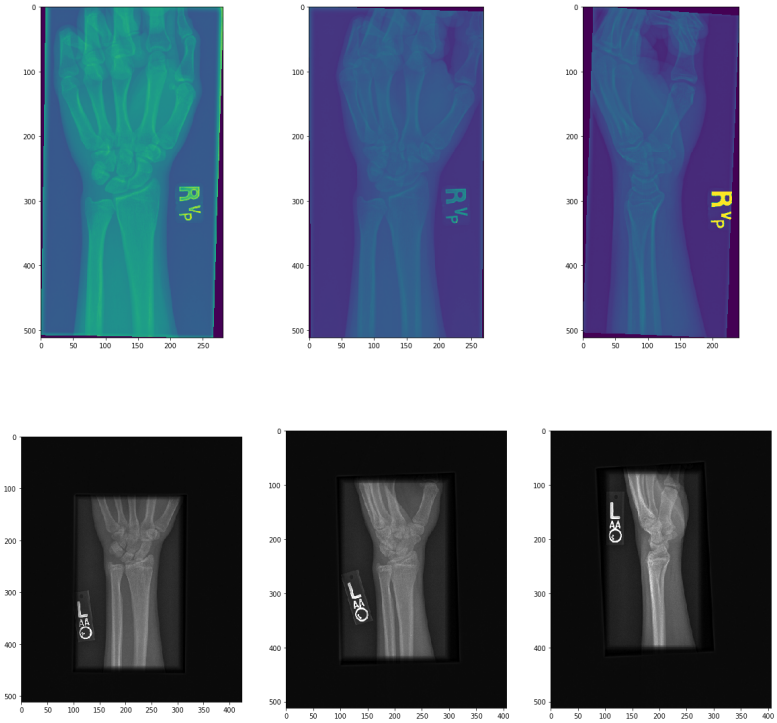
Here we visualize some samples of the data to get more intuition about it.

- We deduced that some data are colored and others are RGB all with different lighting conditions
- Abnormality cannot be deduced by the eyes of a non-professional
- Also that the images are not always zoomed into the desired bones but large portions of some images are useless pixels as seen below.

Normal Wrist Data samples :



Abnormal Wrist Data samples :



Methodology

Data Preprocessing

New images CSV file

We created two new CSV files containing paths of individual images inside studies and their corresponding labels because only full studies were labeled and we decided it is more efficient to train the network on individual images.

And then we read those paths and labels pairs into a pandas dataframe to be able to use flow_from_dataframe() keras function and minimize RAM usage by only reading images when they are needed and read them batch by batch.

Image Resize

All read images before being feed into the network are resized to (224,224,3) to be consistent with the Transfer Learning models used.

Normalization

Pixel Values are normalized to be in the range [-1,1] because this make the training more efficient.

Subtracting Imagenet means

All images are rescaled by subtracting imagenet mean image and standard deviation, this is a recommended practice in general when using models trained on imagenet.

Data Augmentation

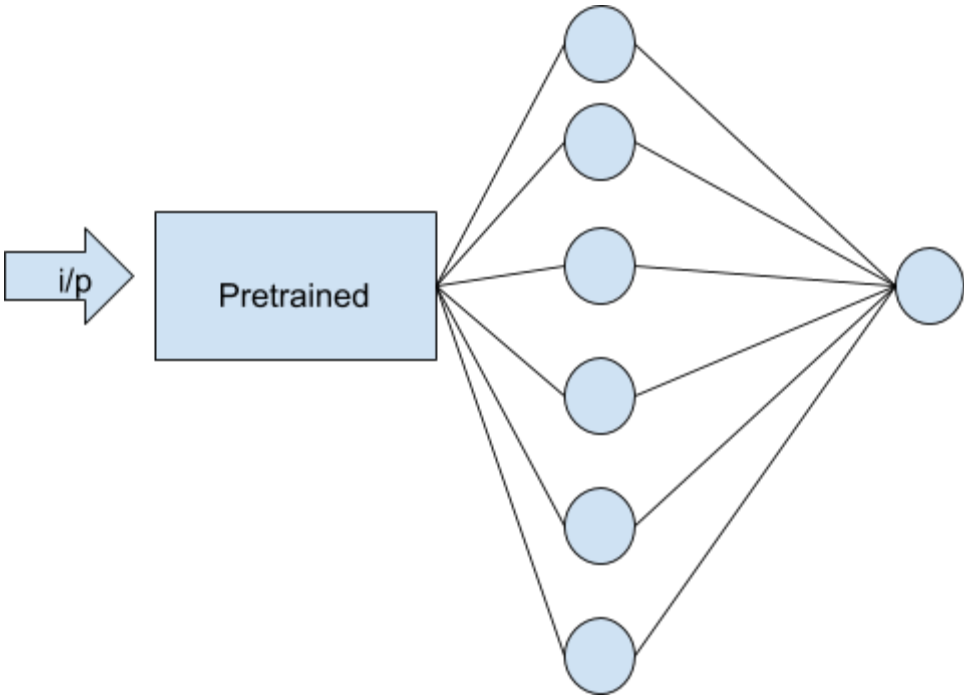
Data is augmented through performing random rotations up to 30 degrees, random horizontal flips and little zoom in and out as recommended by the original paper.

Weighted Binary Cross Entropy

We weighted the binary cross entropy loss function to be less biased due to imbalanced data.

Initial Models

The very basic model used at first was just a Transfer Learning model namely : MobileNet, DenseNet, NASNet Mobile or InceptionV3 followed by one fully connected layer of 128 Neurons with ReLu activation followed by an output Layer with sigmoid activation.

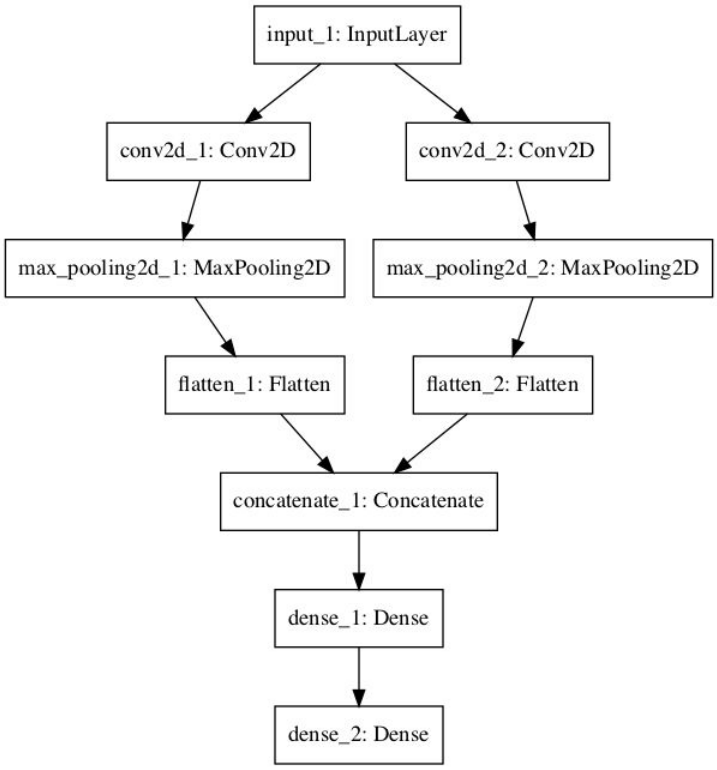


We found that using imagenet weights and freezing nothing at all give the best results.

Feature Engineering

The second type of models used was one aiming to improve features feeded into the final fully connected layers through getting features from several pretrained models and concatenating them then feeding the result into the final fully connected layers.

The figure below shows the idea through normal convolutional layers where conv2D represents a whole pretrained model.

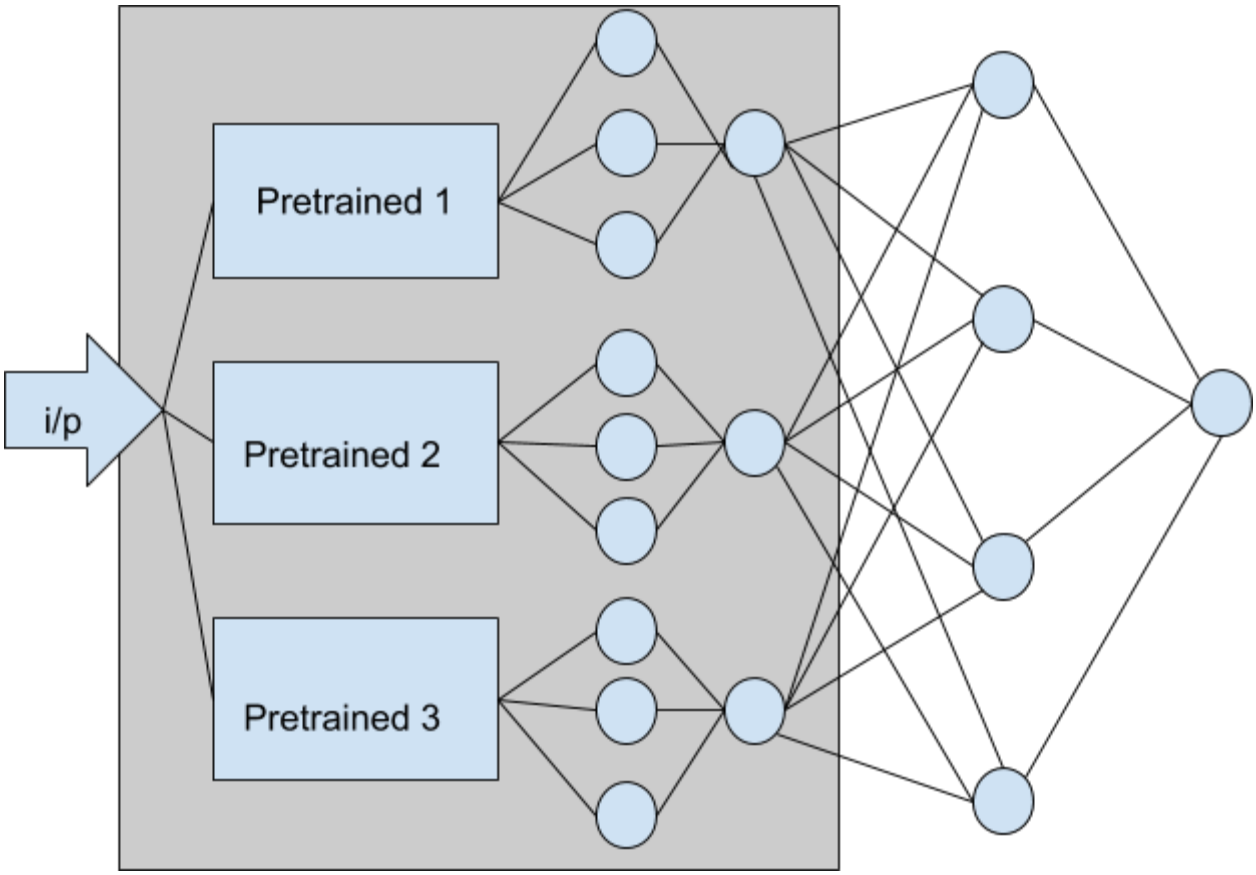


Ensemble Model

The third type of models used is an ensemble model which combines the outputs of full models of the first type i.e. the initial model and feeds those outputs into a fully connected layer followed by an output layer like the ones in the first example.

This is just one type of ensemble models called Stacked Generalization Ensemble but there are many other ways to do them.

The figure below illustrates the general idea behind Stacked Generalization Ensemble.



Extra Features

- Automatic uploading of saved models into Dropbox

Process Complications

- A bug in Keras's flow_from_dataframe function that had cost us alot of time in debugging until we figured it was an issue with the function itself and contacted them until they fixed the issue
- Inability to know whether we should make a model of each limb or one make one model to predict abnormality anytype of limbs which made us try both and choose the one that foreshadows better results
- Google Colab's limited resources with respect to data size and needs which forced us to either implement our own data generator or wait until flow_from_dataframe was fixed and waste a lot of time in trails with small portions of data

- Model overfitting : were initial models in the first five days tended to predict all data as one class and the issue was resolved later.
- We were not very familiar with dealing with such large datasets in size and imbalanced datasets issue and this took time to handle

Results

Basic Models :

NASNet gave 78% training accuracy for training and 72% for validation data which is above benchmark in basic models.

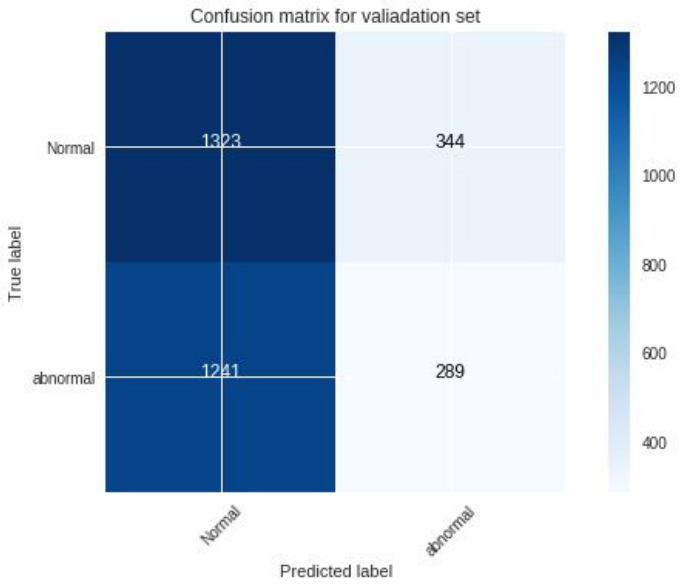
Mobile net Gave 72% in both training and validation .

Both of results above show that there is no overfitting because they exceeded the benchmarks stated above by significant amounts.

But confusion matrices were not produced with those results, and were not included here due to time shortage (but model weights are saved and results can be reproduced).

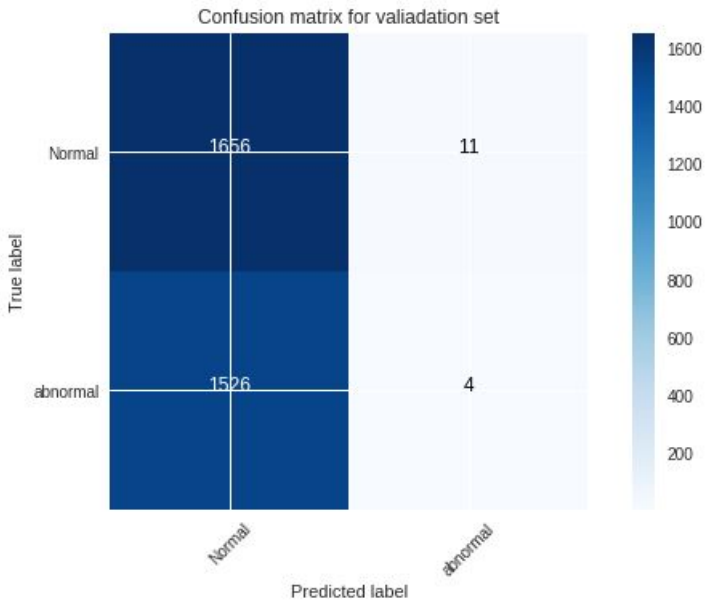
Feature Engineering Models

75% Training and 72% Validation accuracies, validation set confusion matrix:



Ensemble Models

71% Training and 58% Validation accuracies, validation set confusion matrix:



Conclusion

- Imbalanced datasets are very prone to overfitting and biased results to majority class even with data augmentation and weighting loss functions
- Confusion Matrices are better metrics than accuracies in imbalanced data problems