**Deep Learning Solution to Detect Diabetic Retinopathy**

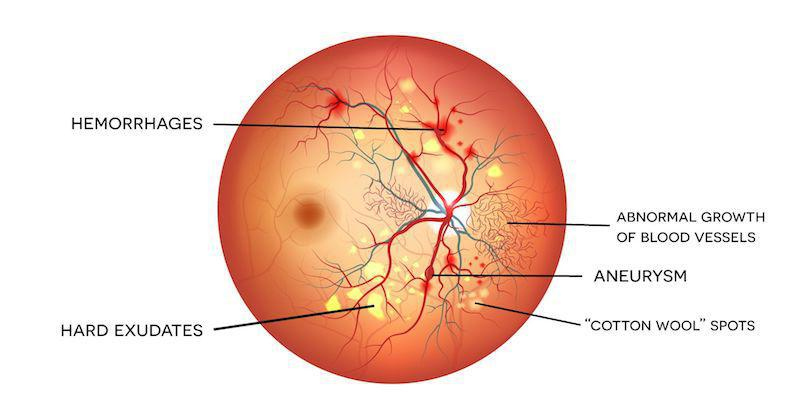
**Executive Summary**

Diabetic Retinopathy is an eye condition which leads to loss of vision and permanent blindness in people having diabetes.This affects the blood vessels in the eye.Our goal is to propose a deep learning solution to detect this disease at an early stage, such that it doesn't cause any permanent damage. This can be deployed in rural areas where conducting medical screening is difficult. And a deep learning solution will be faster in giving out results.

**Problem Description**

Millions of people suffer fromdiabetic retinopathy, the leading cause of blindness among working aged adults. Aravind Eye Hospital in India hopes to detect and prevent this disease among people living in rural areas where medical screening is difficult to conduct.

Diabetic Retinopathy affects the blood vessels in the eye. Based on quick review the following symptoms can be observed in images of eyes.



This can be Hemorrhages, Hard Exudates, Abnormal Growth of Blood Vessels, Aneurysm, and Cotton Wool Spots. The whole eye condition can be categorised into five categories: No Diabetic Retinopathy, Mild Condition, Medium Condition, Severe Condition and Proliferative Diabetic Retinopathy.

Traditionally, technicians travelled to the rural and urban areas to capture images(fundus photography) and then rely on highly trained doctors to review the images and provide diagnosis.

It was time taking and not an efficient method.

Currently as technology has advanced so much and it has made a direct impact on our lives. Now

Instead of relying on doctors to review the images and identifying the issue, we can use a deep learning model which will determine the disease and its stages.

**Resources Utilised and Deep Learning Solution Evaluation**

We are going to use the dataset featured in the APTOS 2019 Blindness Detection Challenge in Kaggle, which is a platform for Machine Learning and Deep learning challenges. And also our deep learning model will be evaluated through the challenge’s public and private datasets, where private dataset holds for 85% of test data and public dataset holds for 15% of test data.

Reference Link: <https://www.kaggle.com/c/aptos2019-blindness-detection/overview>.

**Deep Learning Solution**

We are going to make use of a pretrained model called EfficienetB7 as our base model which is proposed by Google AI, and has established itself as the goto deep learning solution for image classification as it has outperformed many deep learning architectures when trained on the standard imagenet dataset.

We have made use of keras as our front end library and tensorflow as our backend library in python for creating the model and implemented on Kaggle’s Online Jupyter Notebook. And the Training has been done using Kaggle’s TPU accelerator, which has been recently introduced by Kaggle with 30hrs of execution time per week. We have four notebooks, one for preparing the training and validation dataset, training the model, one for submission and for model performance statistics.

**Key Characteristics**

The high performance is obtained because of model scaling, with correct and proper increment in width, height and depth of the images when using convolution and pooling based on mathematical scaling coefficients.

**Key Approach**

**Notebook1: Preparing the Training and Validation Dataset**

We split the dataset to train and cross validation, as validation dataset was not given explicitly. And an observation was made that the dataset provided to us by kaggle was unbalanced, and hence this would cause the model to underperform when the images were of the minority categories.

|  |  |
| --- | --- |
| Diagnosis | Number of Images |
| 0 | 1721 |
| 1 | 349 |
| 2 | 950 |
| 3 | 182 |
| 4 | 276 |

So we decided to oversample the train dataset making the distribution balanced, with each category having an equal number of images. And information about the newly created dataset was stored in the CSV file, which later can be invoked in Model Training Notebook.

**Notebook2: Model Training**

The preprocessing part of the images, we currently had increased the contrast and brightness of the images, in order to make the images clear, but more work has to be done on that. We added a few code snippets to invoke the tpu and convert the dataset to tensorflow’s dataset format for obtaining a faster input pipeline. The model was performed in a sequence of two as kaggle’s limited time execution duration. And initial weights were of the weights of the pretrained model itself, when trained on the imagenet dataset by Google. In the first sequence we trained the model for 10 Epochs without augmentation and saved the weights and used these weights to load in the new version of the notebook. In this notebook we actually trained the data with augmentation using the weights that we actually saved in the previous version (i.e., weights without augmentation) and trained over the new augmented data for 10 Epochs and saved the weights. The training accuracy after all this training we were able to achieve 0.9767 and on the validation dataset we were able to achieve a validation\_accuracy of 0.8696.

**Notebook3: Submission**

As part of the compeitation rules the notebook we are submitting should not be connected to the internet and hence, we added a python installation file for installing the Efficient B7 model. So that the notebook doesn’t have to download it from the internet. Applied the same preprocessing on the images as we had trained, and predicted results on the test dataset, and saved the results to the submission.csv file and made the submission.

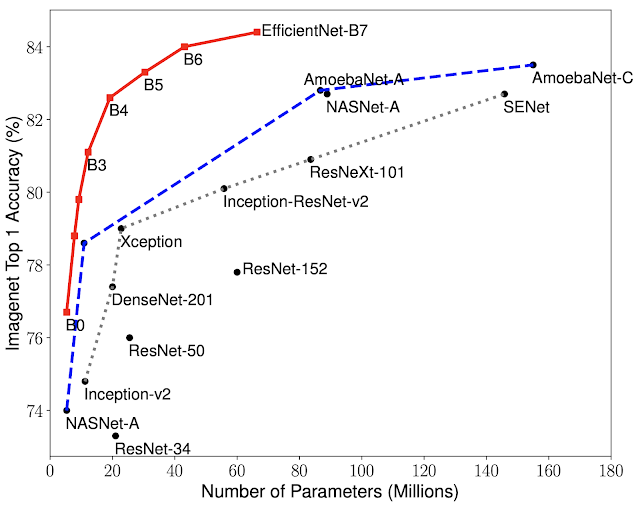
**Notebook4: Model Performance Statistics**

This notebook displays plots and extra information regarding the model’s performance.

**Different methodologies used**

At first we used a pretrained model called ‘densenet’ but the trainable parameters were low.

So we made use of another pretrained model called ‘EfficienetB7’ which was having higher performance than the densenet.



Without Oversampling: Even though when we see the general accuracy was good in the baseline model’s approach, if we examine the performance of the model in each category, the model was not doing well in the underrepresented categories.

Oversampling: Oversampling the underrepresented categories, helps the model in paying the same amount of attention as the major category.

With Augmentation and Oversampling: Because oversampling resulted in overfitting, inorder to introduce some amount of uniqueness in each images, we applied augmentation techniques like random rotation, random brightness, random hue, random contrast and gamma adjustment.

With Preprocessing: As the images were not that bright and not very smooth, the brightness, contrast,etc of the images were adjusted.

Dropout: In order to reduce overfitting that was observed, we added a dropout layer, which made the model dropout or turnoff a few nodes in the layer, so as to reduce overfitting in the training dataset.

Usage of various Models: Used models like ResNet, DenseNet 201, but optimal performance in the current model.

**Results of various techniques**

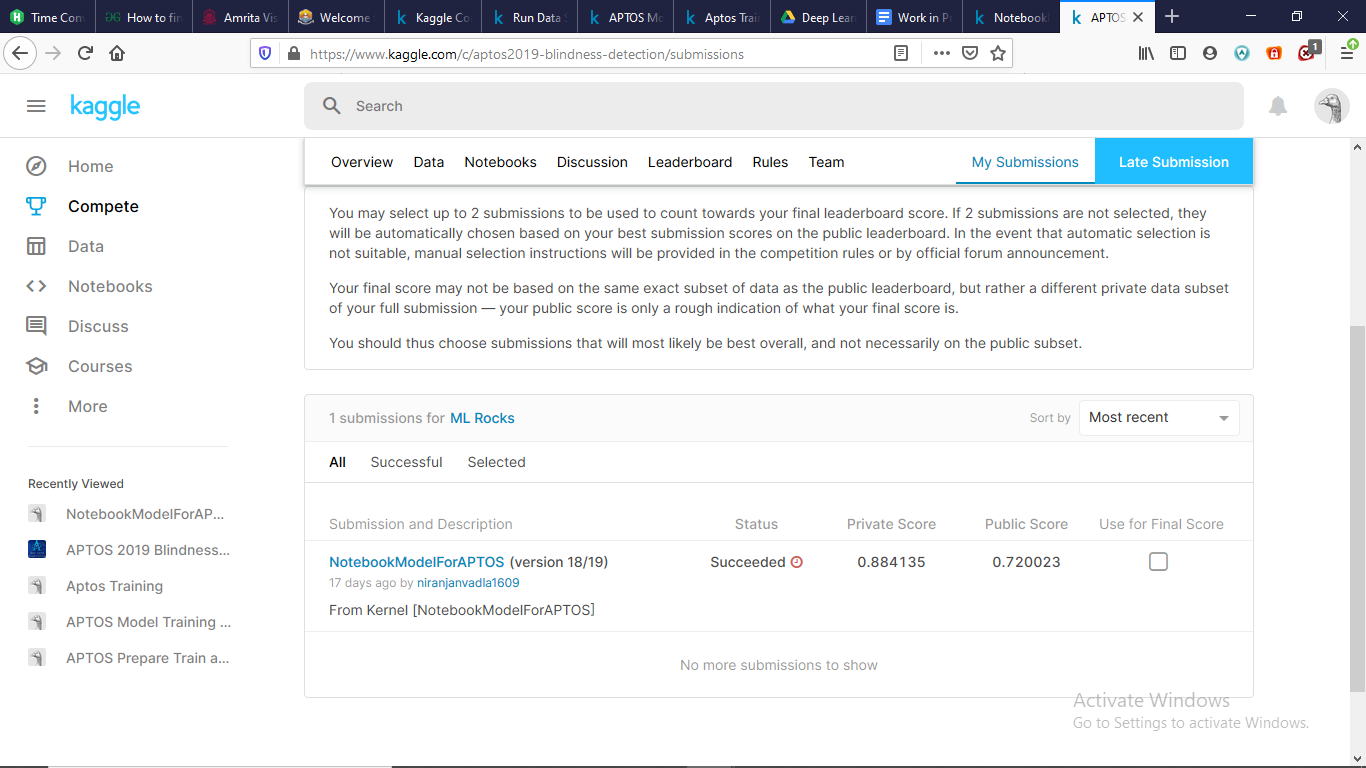
1. Without oversampling - Model was underfitting, because of unbalance in the dataset.
2. Oversampling - Underfitting changed to Overfitting.
3. With augmentation and oversampling - Optimal Performance was obtained.
4. With preprocessing - A slight increment in the performance was observed, as images became a bit clear
5. Used Dropout - Model was underfitting.
6. Used Various Models like ResNet, DenseNet201 to choose the best model - Currently selected model shows better performance.

**Conclusion**

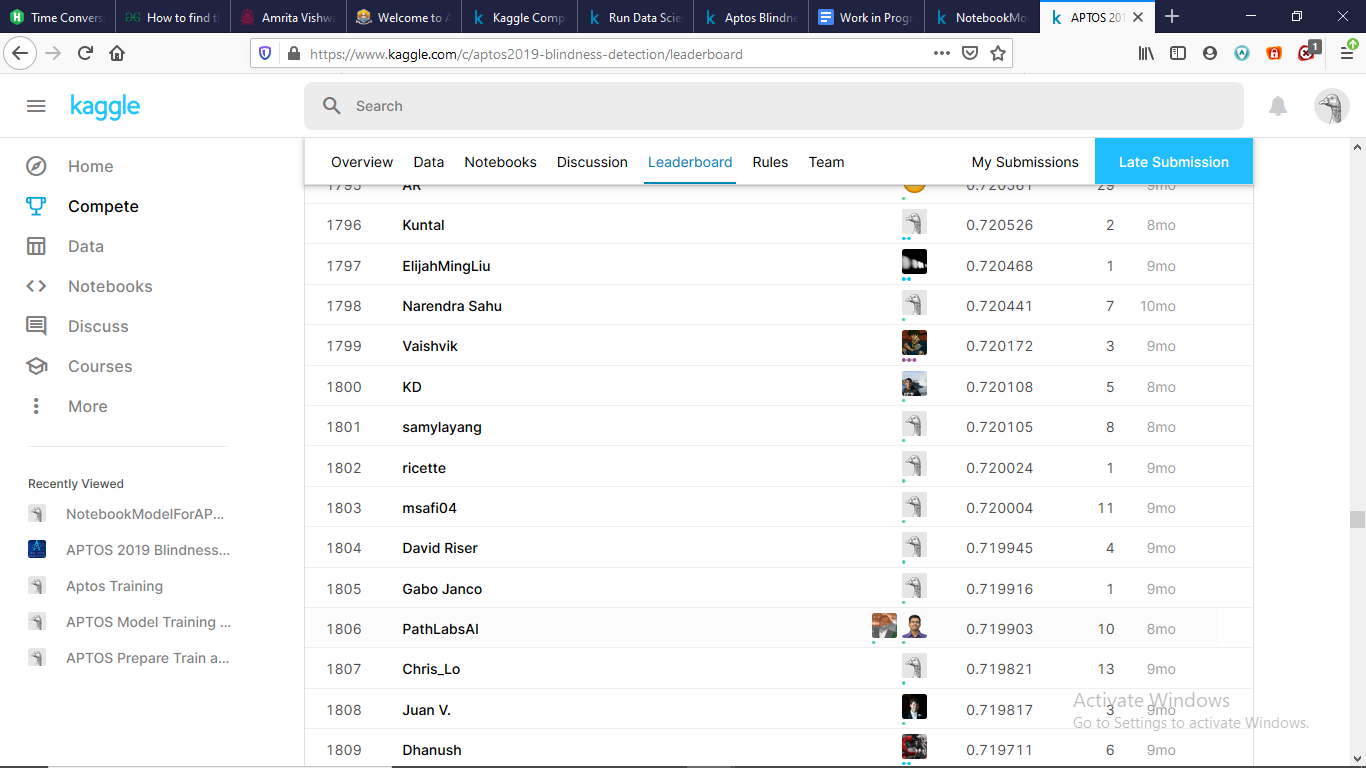
**Scores Obtained**

As of now the best score obtained was 0.884135 on the private dataset(85% of test data) and 0.720023 on public dataset(15% test data). The highest score on the private dataset is 0.936129 and on the public dataset is 0.856139.

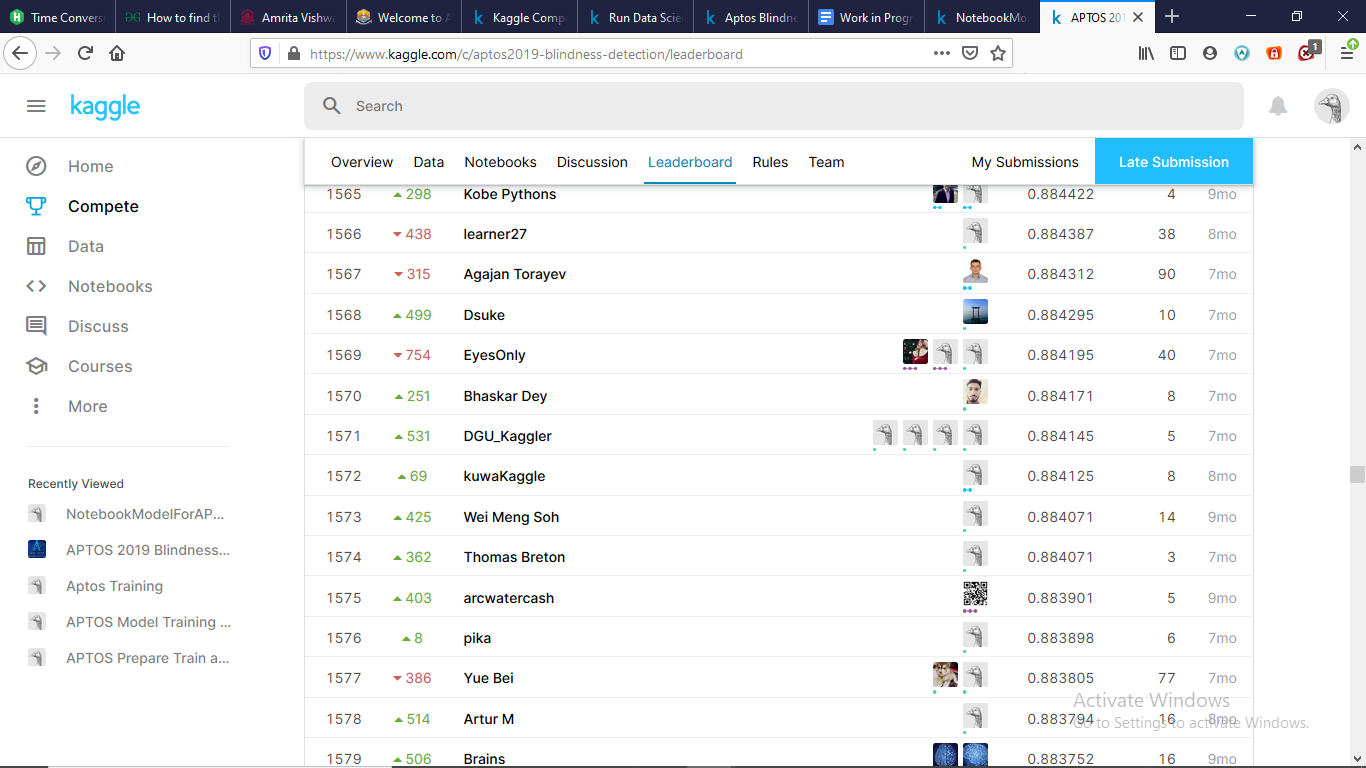
These are our private and public scores



This is where our public score lies



This is where our private score lies



By preparing this model we are trying to contribute towards the betterment of humanity by detecting the disease at an early stage and come up with a model that can result in greater accuracy. This results in fast screening over a large number of people in a short amount of time and we can help people specifically in the rural areas where they have very less access to these kinds of facilities.