Machine Learning Engineer Nanodegree Program

Report

STEM NGUYEN

Blindness Detection (Kaggle Competition)

Project Overview

Imagine being able to detect blindness before it happened. Millions of people suffer from diabetic 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. Successful entries in this competition will improve the hospital's ability to identify potential patients. Further, the solutions will be spread to other Ophthalmologists through the 4th Asia Pacific Tele-Ophthalmology Society (APTOS) Symposium.

Currently, Aravind technicians travel to these rural areas to capture images and then rely on highly trained doctors to review the images and provide diagnosis. Their goal is to scale their efforts through technology; to gain the ability to automatically screen images for disease and provide information on how severe the condition may be.

References
https://www.technology.org/2019/07/15/aptos-2019-blindness-detection/
https://www.kaggle.com/c/aptos2019-blindness-detection
https://towardsdatascience.com/aptos-2019-blindness-detection-520ae2a4acc

Problem Statement

This is a deep learning problem. Inputs are the images and the goal is to predict severity of diabetic retinopathy on a scale of 0 to 4:
0 - No DR
1 - Mild
2 - Moderate
3 - Severe
4 - Proliferative DR

Metrics

Precision
Let's start with precision, which answers the following question: what proportion of predicted Positives is truly Positive?

Precision = (TP)/(TP+FP)
In the asteroid prediction problem, we never predicted a true positive.
And thus precision=0

Recall

Another very useful measure is recall, which answers a different question: what proportion of actual Positives is correctly classified?

Recall = (TP)/(TP+FN)

In the asteroid prediction problem, we never predicted a true positive.

And thus recall is also equal to 0.

F1 Score

The F1 Score is the 2*((precision*recall)/(precision+recall)). It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.

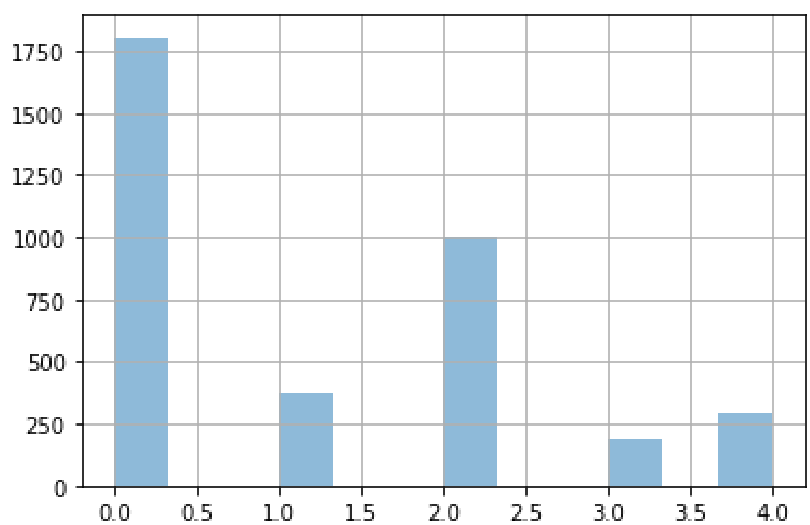
Analysis

Data Exploration

Datasets are **3662** color images with png format.

Example:

Dataset is imbalanced



We have 4 types of images.

0 - No DR

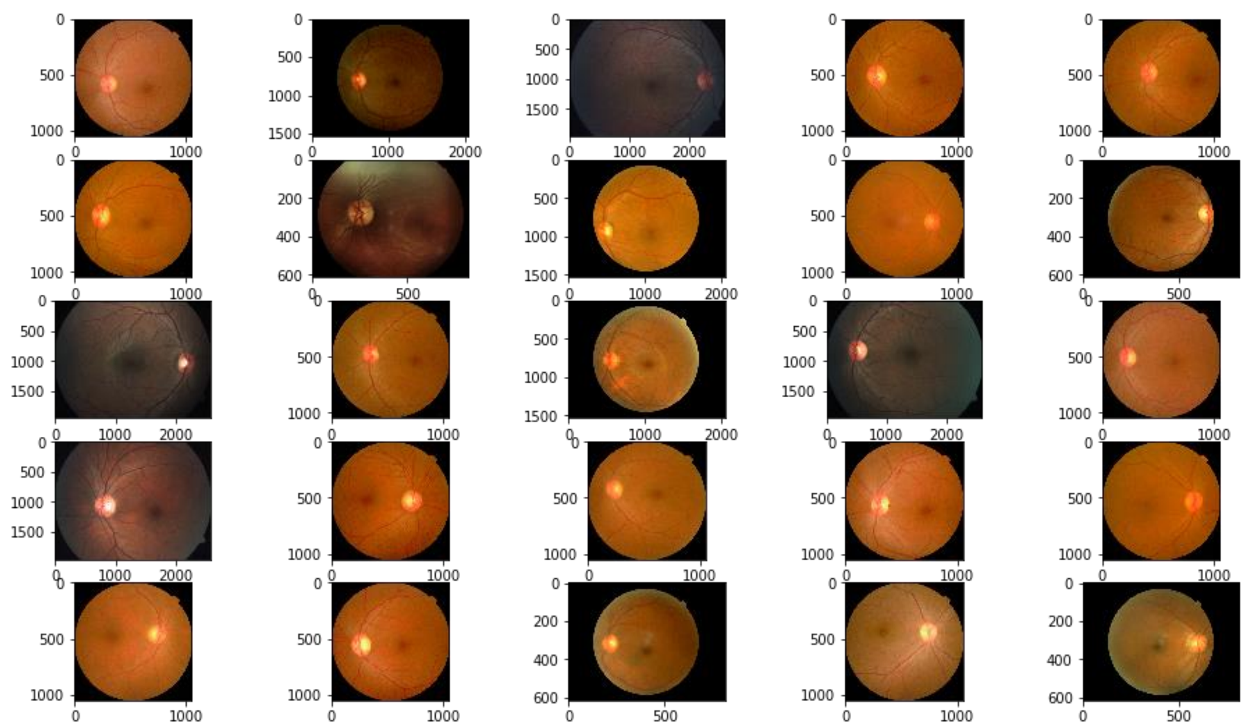
1 - Mild

2 - Moderate

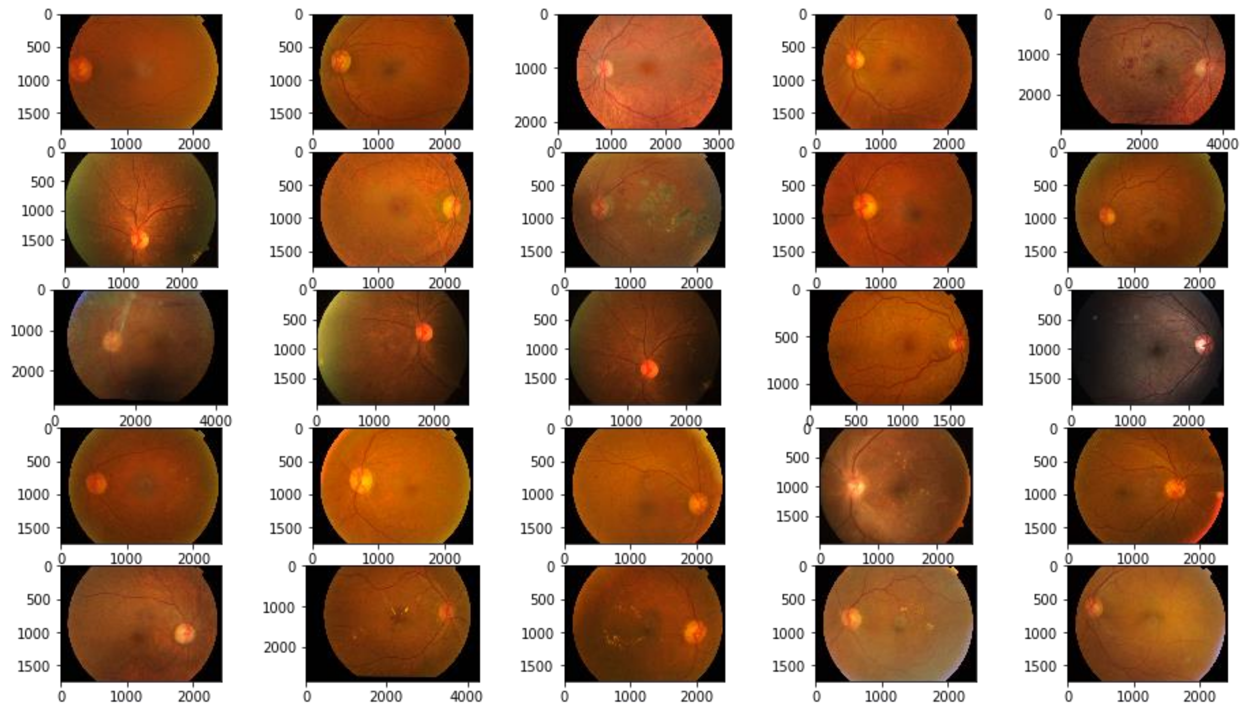
3 - Severe

4 - Proliferative DR

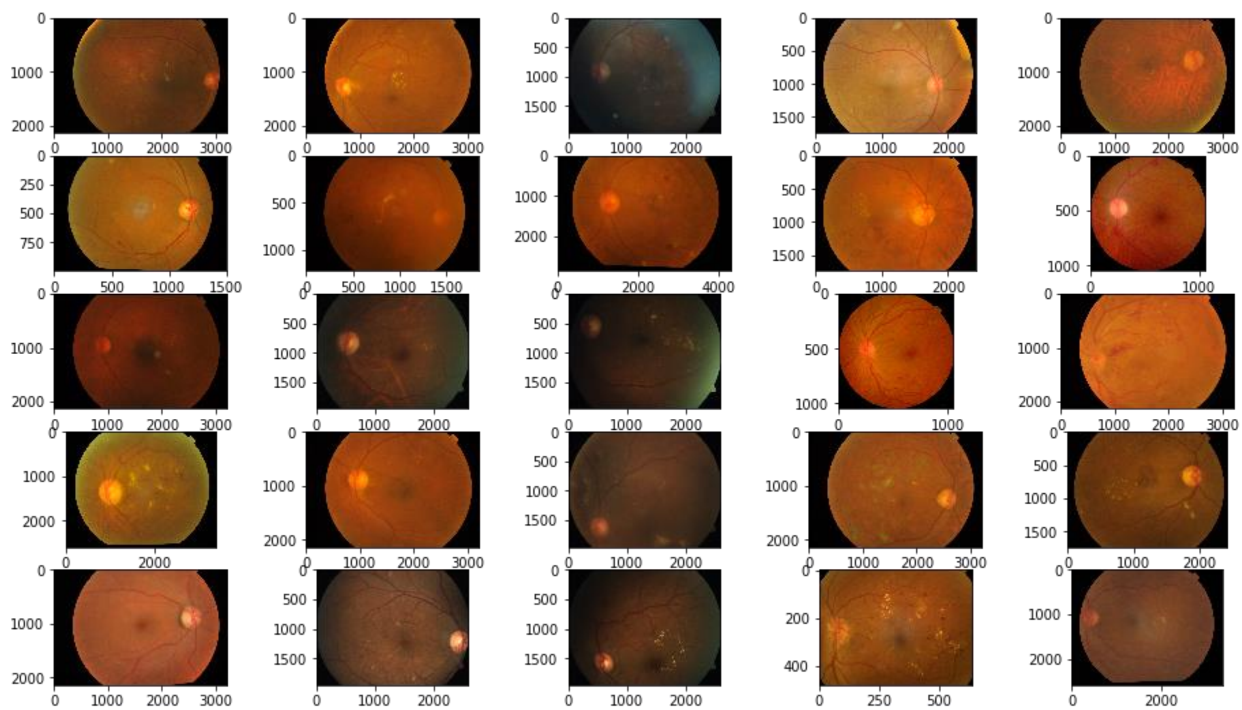
0 - No DR



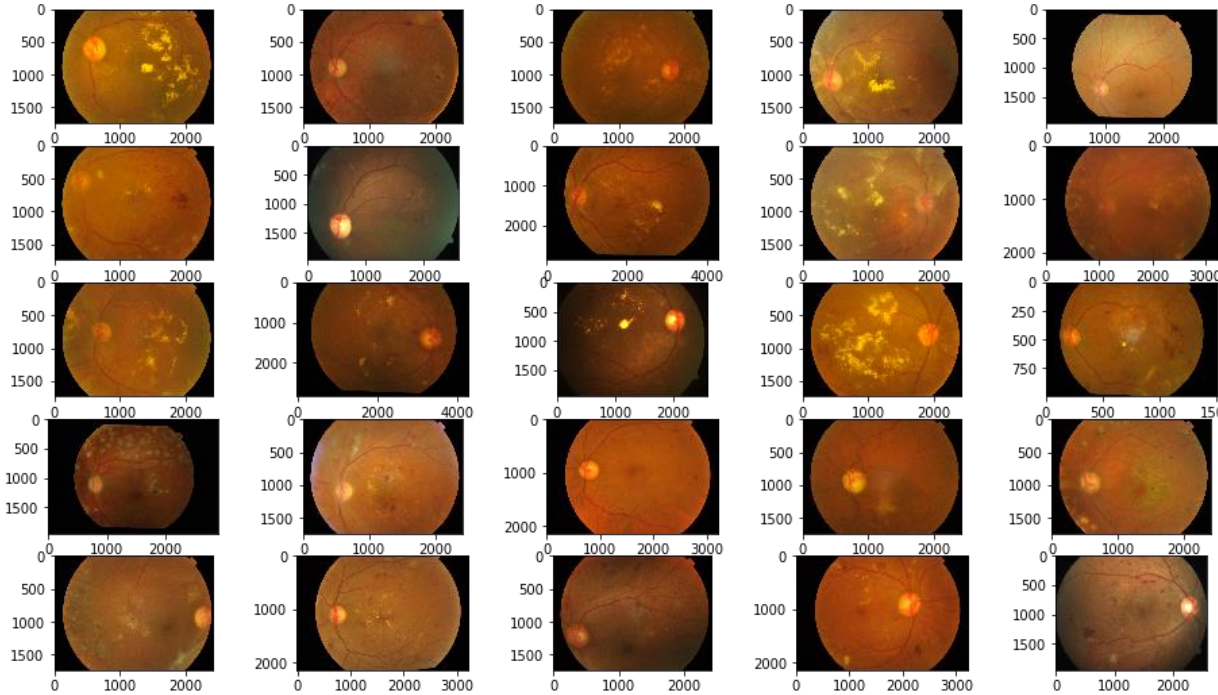
Severity 1: Mild



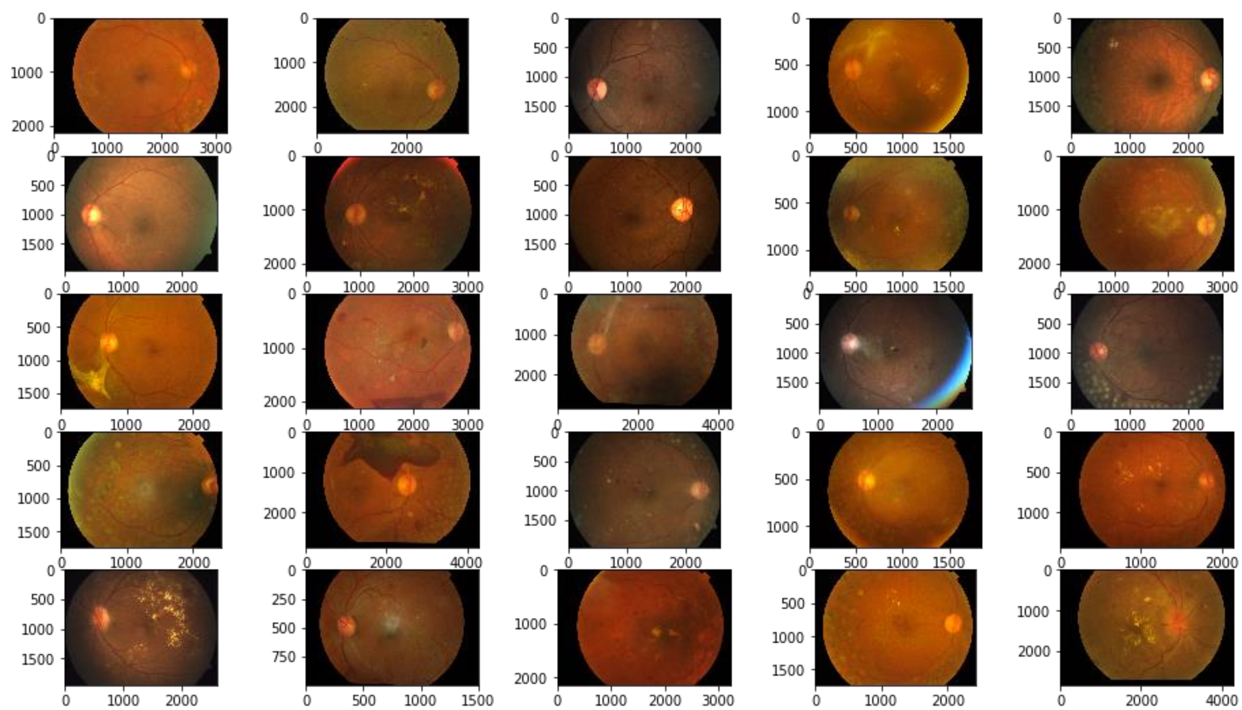
Severity 2: Moderate



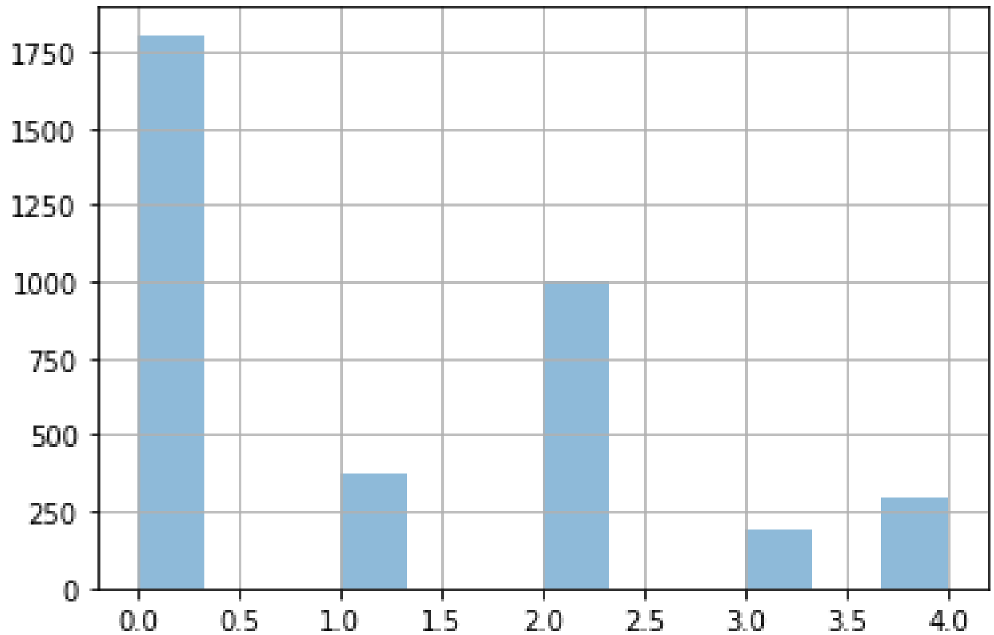
Severity 3: Severe



Severity 4: Proliferative DR



Exploratory Visualization



So, we can see the data is imbalanced.

Algorithms and Techniques

Resnet

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. An additional weight matrix may be used to learn the skip weights; these models are known as HighwayNets. Models with several parallel skips are referred to as DenseNets. In the context of residual neural networks, a non-residual network may be described as a plain network.

SeResnet

Base class for SE-ResNet architecture.

This architecture is based on ResNet. A squeeze-and-excitation block is applied at the end of each non-identity branch of residual block. Please refer to the original paper for a detailed description of network architecture.

In this machine learning problem, we used 3 models in Resnet ('resnet18', 'resnet34', 'resnet50') and 1 model in SeResnet ('seresnext50').

Methodology

Data Preprocessing

Resize the images to 224x224 for inputs. Change the color from BGR to RGB. Scaling and Normalization operations on data and splitting the data in training, validation and testing sets.

I used ing library to load images

im = ing.imread(image_path)

cv2 library to transfer color of the images

im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)

and cv2 library to resize the images

im = cv2.resize(im, (desired_size, desired_size))

Implementation

Two main stages: training, test.

I defined the method like below

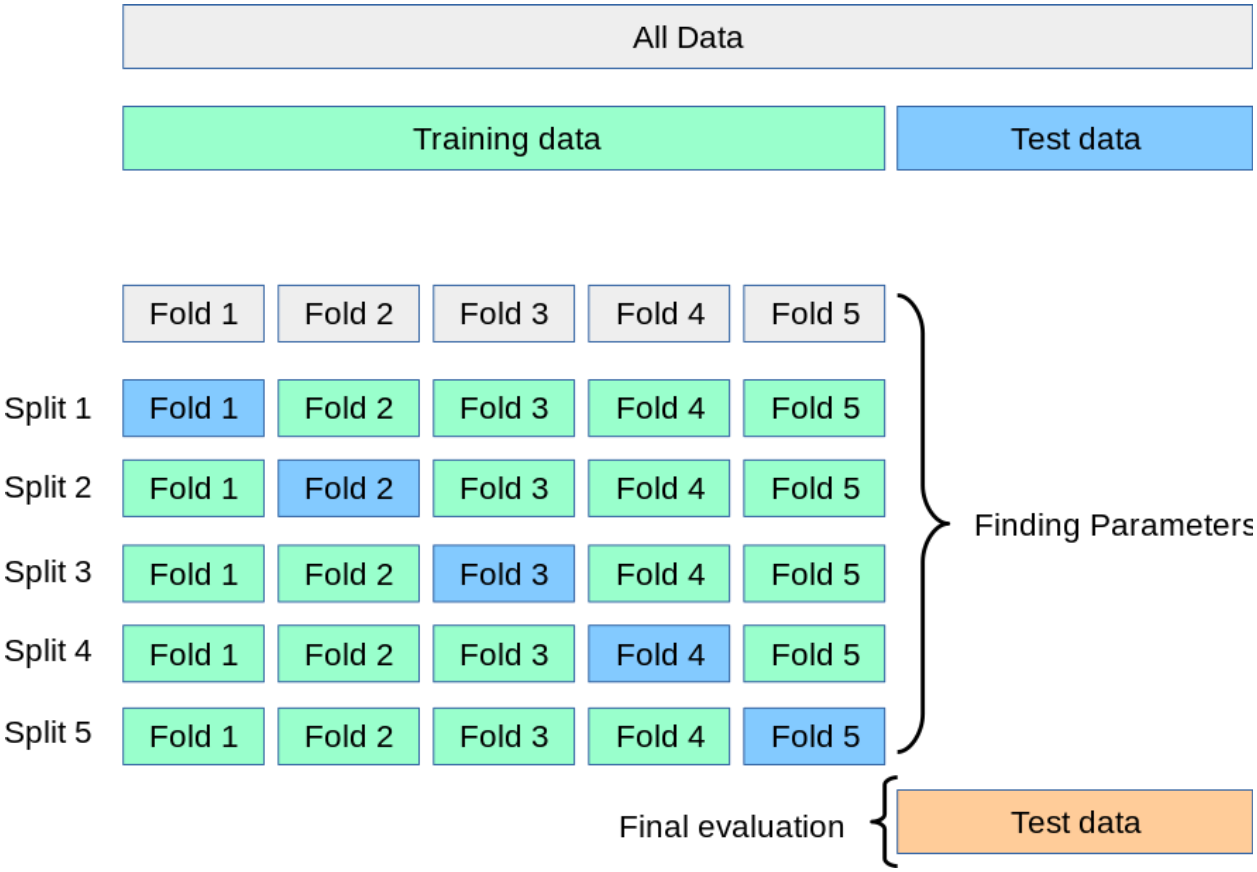
def train_model_by_name(model_name ='resnet18')

I did the for loop to train one by one model based on model name.

['resnet18', 'seresnext50', 'resnet34', 'resnet50']

When I train to model, I used kfold.

n_splits=5; that's mean 80% data for training and 20% for testing. We repeat that 5 times like below.



I used the library classification_models from https://github.com/qubvel/classification_models to train the models. I train with 4 model names 'resnet18', 'seresnext50', 'resnet34', 'resnet50'.

After training, I have the precision, recall, f1 score for each model like below.

```
{
 'resnet18': [0.795483, 0.75918734, 0.77659225],
 'seresnext50': [0.8106526, 0.7914171, 0.8007201],
 'resnet34': [0.78417337, 0.7609518, 0.7721284],
 'resnet50': [0.7330123, 0.68673646, 0.70864904]
}
```

We can see the seresnext50 model which has the highest score.

Results

Benchmark

Solutions	Accuration	Data
resnet18	0.77659225 (Cross validation 5 folds)	3662 images
seresnext50	0.8007201 (Cross validation 5 folds)	3662 images
resnet34	0.7721284 (Cross validation 5 folds)	3662 images
resnet50	0.70864904 (Cross validation 5 folds)	3662 images

After training for 4 models. The scores like below.

resnet18: 0.77659225
seresnext50: 0.8007201
resnet34: 0.7721284
resnet50: 0.70864904

Based on the results above, I will choose the model seresnext50 for this problem.

The accuracy is 0.8 (80%) is acceptable. Because "National Institutes of Health's recommended standard of at least 80 percent accuracy and precision for diabetic retinopathy screens." – Reference https://www.wired.com/2016/11/googles-ai-reads-retinas-prevent-blindness-diabetics/