Machine Learning Engineer Nanodegree

Capstone Proposal

Karry Harsh
July 23rd , 2019

Proposal

Blindness Detection(Kaggle Completion).

Domain Background

We are are familiar with **Diabetes mellitus (DM),** commonly known as **diabetes**, is a group of metabolic disorders characterized by high blood sugar levels over a prolonged period. according to IDF(International Diabetes Federation) 425 Million adults have diabetes in which 1 in 2 remain undiagnosed. One of many causes of it is Diabetic retinopathy eye Disease.

Diabetic retinopathy affects blood vessels in the light-sensitive tissue called the retina that lines the back of the eye. It is the most common cause of vision loss among people with diabetes and the leading cause of vision impairment and blindness among working-age adults. At least 90% of new cases could be reduced with proper treatment and monitoring of the eyes.

Problem Statement

The goal of the project is to detect and prevent this disease by predicting patient Condition and in which stage the patient is in. I will be tackling this problem using Deep neural network Image classification technique to classify the severity of diabetic retinopathy. Doing it so, make it easier to detect and prevent this disease among people living in rural areas where medical screening is difficult to conduct.

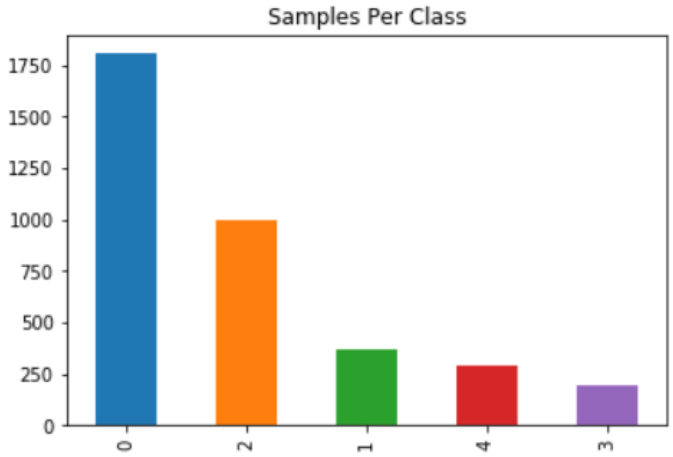
Datasets and inputs

The datasets are large set of retina images taken using fundus photography under a variety of imaging conditions provided on Kaggle competition website.

A clinician has rated each image for the severity of diabetic retinopathy on a scale of 0 to 4:

0 - No DR
1 - Mild
2 - Moderate
3 - Severe
4 - Proliferative DR

Total Images – Training Data – 3663
Test data – 1929



As we can see there is a class imbalance in the dataset and size of images will also varies, to overcome that sampling of data is required and resizing the image to a standard shape is also necessary.

The dataset contains:

- train.csv - the training labels with columns id_code and diagnosis.
- test.csv - the test set for each id_code a predicted diagnosis to be mapped.
- sample_submission.csv - a sample submission file in the correct format.
- train.zip - the training set images.
- test.zip - the public test set images.

As this is a kaggle competition. The dataset is provided by kaggle can be obtained here

Solution Statement

The solution will be the classification of image using Deep learning CNN as it is very good in pattern recognition. Initially I plan to do some visualization of data to get some understanding. Then, pre-processing of the data (like image augmentation) to make it ready for training in the CNN model. I may use pre-trained CNN to extract feature from image and then use transfer learning to train my model on the given data set. I plan to spend 60% of the time on Pre processing and CNN model selection part and 40% of the time on training models and tweaking parameters.

Benchmark Model

As this is a kaggle competition, a benchmark model would be the best kaggle score for the test set. This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters). In the event that there is less agreement between the raters than expected by chance, this metric may go below 0. However, due to hassle and limitations (like 5 entries per day), for academic purpose, I would use a part of training data as testing data.

I will train my model with different CNN and optimize them and check with which CNN model provides me better score and for satisfying the curiosity, run it on test set provided by kaggle website to check the score. Then, I can compare my model with the other participant model as a benchmark model.

Evaluation Metrics

Since the official evaluation of this project is done by kaggle using quadratic weighted kappa, which measures the agreement between two ratings. This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters). In the event that there is less agreement between the raters than expected by chance, this metric may go below 0. The quadratic weighted kappa is calculated between the scores assigned by the human rater and the predicted scores.

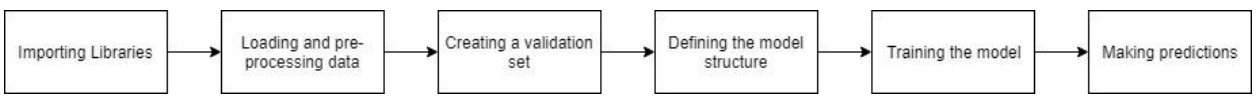
Images have five possible ratings, 0,1,2,3,4. Each image is characterized by a tuple *(e,e)*, which corresponds to its scores by *Rater A* (human) and *Rater B* (predicted). The quadratic weighted kappa is calculated as follows: First, an N x N histogram matrix *O* is constructed, such that *O* corresponds to the number of images that received a rating *i* by *A* and a rating *j* by *B*. An *N-by-N* matrix of weights, *w*, is calculated based on the difference between raters' scores.

An N-by-N histogram matrix of expected ratings, E, is calculated, assuming that there is no correlation between rating scores. This is calculated as the outer product between the actual rating's histogram vector of ratings and the predicted rating's histogram vector of ratings, normalized such that E and O have the same sum.

From these three matrices, the quadratic weighted kappa is calculated.

Project Design

Structure of our image data is already mentioned in Dataset and input section. Initially I will load and pre-process the data and visualize the data for better understanding then I will split my training data into training set to train my model, validation set to see how my data performs on unseen data and then use test data set to for making predictions. After that definition of the model architecture will take place where will try various combination to define an optimize model and tune hyper parameters to decide how good the predictions will be. Then training of the model takes place where training images and their corresponding true label gets train and then validates the model using validation image and their corresponding true label under no. of epochs defined and Finally will estimate the model performance using test images.



Reference

- Kaggle
- IDF
- National eye Institute
- Analytics Vidhya