Machine Learning Engineer Nanodegree

Capstone Proposal

M P S Vishal Singh Tomar

10 September 2019

Proposal

Detecting diabetic retinopathy :

Detect diabetic retinopathy to stop blindness before it’s too late

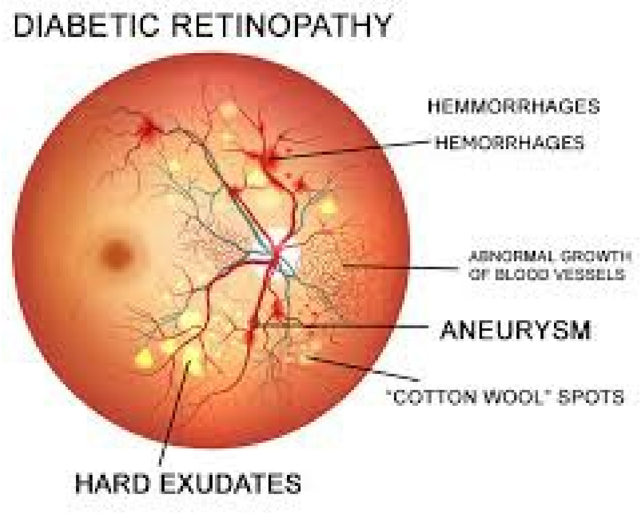
Domain Background

Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.
I want to leverage the power of computer vision and build something that can aid help in the field of health care. This project can help achieve this goal by using image processing.

Problem Statement

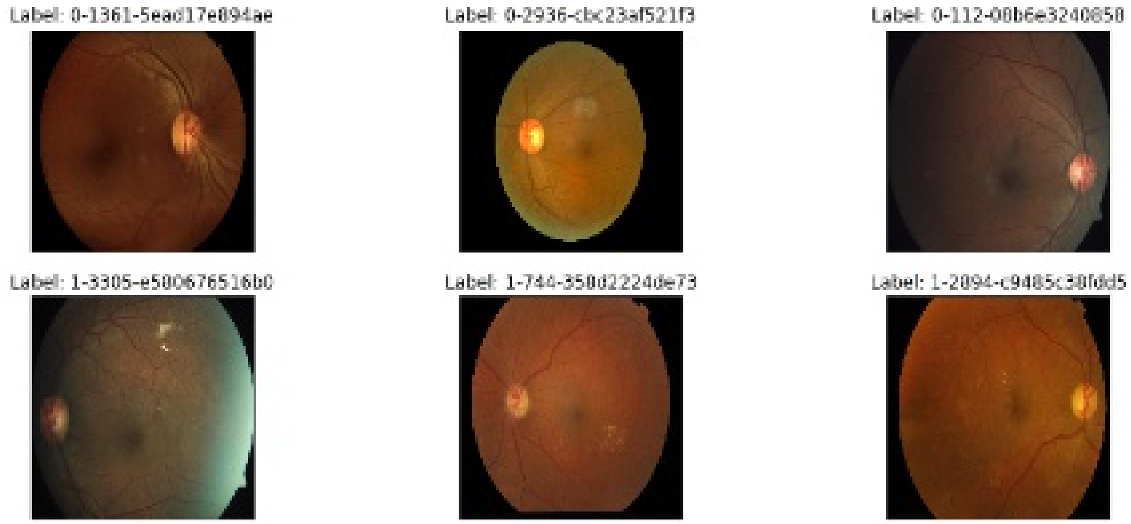
Diabetic retinopathy (DR) is the fastest growing cause of blindness, with nearly **415 million** diabetic patients at risk worldwide. If caught early, the disease can be treated; if not, it can lead to irreversible blindness.

Lets Look at the following diagram to have a better understanding of the problem.



- There are at least 5 things to spot on.
- These five spots are not easily detectable.
- infact some images in dataset have very poor lighting which makes it even more harder to spot these spots. These things make tarining a deep Learning model difficult.

The following figure is a screen shot of dataset images.


Label: 0-1361-5ead17e894ae
Label: 0-2936-cbc23af521f3
Label: 0-112-08b6e3240858
Label: 1-3305-e5006765 16b0
Label: 1-744-350d2224de73
Label: 1-2894-c9485c38fdd5

It is clear that images have poor lighting and nerves and spots which are to be checked are hardly visible.
We need a way to improve its lighting conditions and aslo highlight nerves,spots and try to remove other not so relevent features from the images so the our model can efficiently perform.

Datasets and Inputs

This data set is taken from kaggle competitions it is publicly available as a part of APTOS 2019 Blindness Detection competition.

Dataset includes both images and csv files.

- csv files named as **train.csv** and **test.csv** contains
    1. id_code
    2. diagnosis

- **id_code** is same as the image file name i.e., Each row represents a image filename as **id_code** and what category it belongs to as **diagnosis**.

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
```

1. Above diagram in problem statment shows image samples in the dataset to be used.
2. They are retinal photographic images.
3. These images indicate damage to eyes by showing blood cloting, darkspots or brusted blood vessels.
4. These things are the root causes for blindness. Hence instead of tracking Patient’s medical record data which will have many irrelevent parameters we can focus on these symptoms.
5. The dataset we use contains images which are cassified into below 5 classes.

6. Number of images per classes are:
    - Class 0 (No DR) : 1805 images (49.2%)
    - Class 1 (Mild) : 370 images (10.1%)
    - Class 2 (Moderate) : 999 images (27.28%)
    - Class 3 (Severe) : 193 images (5.27%)
    - Class 4 (Proliferative DR) : 295 images (8.05%)
7. Hence we can conclude that dataset is largly imbalanced.
8. I will randomly split data into train,test and validation sets with 60%,20% and 20% respectively

Solution Statement

This is a image classification problem like the dog breed classifier. The same technique can be used in the proposed system. I intend to use a deep neural network which was built using transfer learning and is proven succesfull in dog breed classification. Although we will have to tweek the netrok a little bit as dog breeds were already a part of pretrained model which is not the case with ratina images.

As discribed in the problem statment the images are not clear and very hard to understand. we will follow these steps to process the images before feeding them to Neural Network for training.

- First we will rescale the images and remove black background.

- we will try to improve light condition by applying Ben Graham’s preprocessing method[1].

- Then we will try circle cropping to get a nice circular image of the ratina.This will improve the uneven image sizes.

- The next step will be to train a Deep Neural Network with these data sets. we will apply transfer learning using DenseNet pre - trained model. We then add our own layers to adapt the Model to our dataset.

We will use the evaluation metrics described in later sections to compare the performance of
these solutions against the benchmark models in the next section

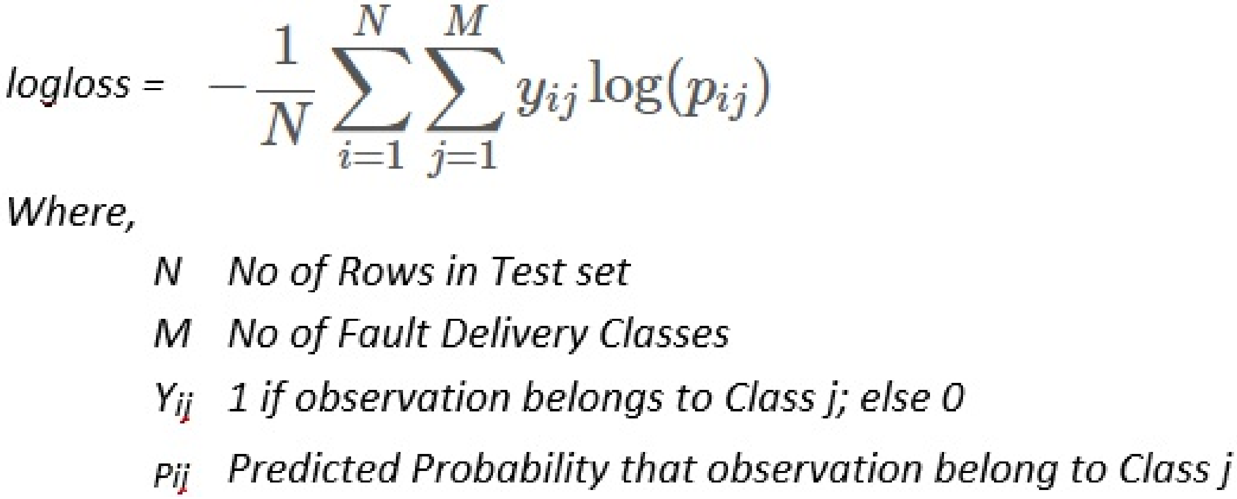
Benchmark Model

This project is still under research hence no solid papres have been published on benchmarks. Hence i would like to use the following notebook published publicly by KeepLearning as benchmark.

- This is a ResNet50 implementation with loss=‘categorical_crossentropy’. Applied on image size = 300.

Evaluation Metrics

I will use Log-loss function for performance analysis. Logarithmic Loss or log Loss, Works by penalising the false classifications. It works well for multi-class classification. When working with Log Loss, the classifier must assign probablity to each class for all the samples . Suppose, there are N samples belonging to M classes, then the Log Loss is calculated as below:



$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

Where,

N *No of Rows in Test set*

M *No of Fault Delivery Classes*

Y_{ij} *1 if observation belongs to Class j; else 0*

P_{ij} *Predicted Probability that observation belong to Class j*

Log Loss has no upper bound and it exists on the range [0, ∞). Log Loss nearer to 0 indicates higher accuracy, whereas if the Log Loss is away from 0 then it indicates lower accuracy.
In general, minimising Log Loss gives greater accuracy for the classifier.

Log Loss is a Micro -average matrix.

Micro-average is preferable if there is a class imbalance problem.

In our sample data we cannot assume to have a balanced set of images for all the classes hence I prefer using Log Loss function.

Project Design

Data Preprocessing

- First we will load the image.
- Resize the images to 224 x 224 so that it matches the ImageNet format.
- Then we will use Ben Graham’s preprocessing method [2] to improve the lightning condition. As mentioned in the above problem statment image clarity is poor due to dull lighting.
    - We will first try to sharpen image using GaussianBlur function.
    - Then crop the image to remove black space on the side.

Mixup & Data Generator

We will create a data generator that will perform random transformation to our datasets (flip vertically/horizontally, rotation, zooming). This will help our model generalize better to the data.

Data Splitting

Split the data into a training set and validation set with an 80-20 split.

Model training and evaluation

- We will use a DenseNet-121 pre-trained on ImageNet.
- we will use ‘binary_crossentropy’ as loss function and ‘accuracy’ as evaluation matrix to train the model.
- We will finetune it using Adam for 15 epochs, and evaluate it .

References

Oxford Applied and Theoretical Machine Learning Group :github.com/OATML/bdl-benchmarks/projects

[2] Ben Graham’s preprocessing method : https://github.com/btgraham/SparseConvNet/tree/kaggle_Diabetic_Retino

[3] DenseNet: https://github.com/liuzhuang13/DenseNet

[4] Log Loss Matrix : https://datawookie.netlify.com/blog/2015/12/making-sense-of-logarithmic-loss/