SOULPAGE
IT SOLUTIONS

Detecting Blindness in Diabetic Retinopathy patients using Deep learning

Detecting Blindness in Diabetic Retinopathy patients

Problem statement:- Detect Diabetic retinopathy to stop blindness before it is too late
Diabetic eye disease is a group of eye conditions that can affect people with diabetes. Every year millions of people are suffering from diabetic retinopathy which is also one major cause to get blind in the early ages of people. In 2014, there were approximately 422 million people (8.5% of the world's adult population) living with diabetes; compared to 108 million in 1980 (2016 WHO Global Report on Diabetes). We hope to detect and prevent this disease among people living in rural areas where medical screening is difficult to conduct. We can help hospitals to identify potential patients, that makes up the process faster in treating the patients who are actually at the final stages of the disease.

Solution:
In this Document, we used Deep learning model with Convolution Neural networks, which are really good at dealing with images. We will be using Resnet101 model to predict the stages or labels in our case such as

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

Where every predicted label says about every stage he/she is in.

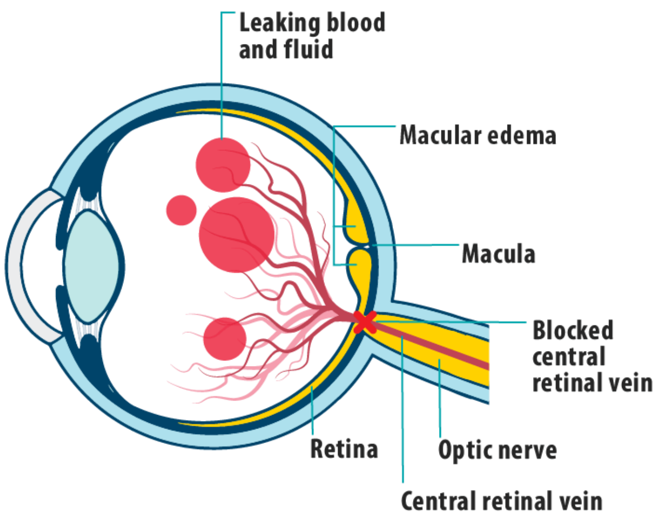
Tools used:-

1. Pandas
2. Numpy
3. Pytorch
4. Deep learning
5. CNN Model

Dataset:-
We have a dataset of 3500 images to train our model and 1900 images to test the trained model, where each and every image is a sample collected from people with Diabetic retinopathy in rural areas of India. The training dataset we have is labelled data with classes like 0,1,2,3 and 4. Collectively we have 10GB of data with images.

Diabetic Retinopathy:-

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. It can also affect many parts of the eye, including the retina, macula, lens, and the optic nerve. Diabetic Retinopathy can also cause problems like Cataract and Glaucoma. In future, this work can also be extended in identifying Cataract and Glaucoma.

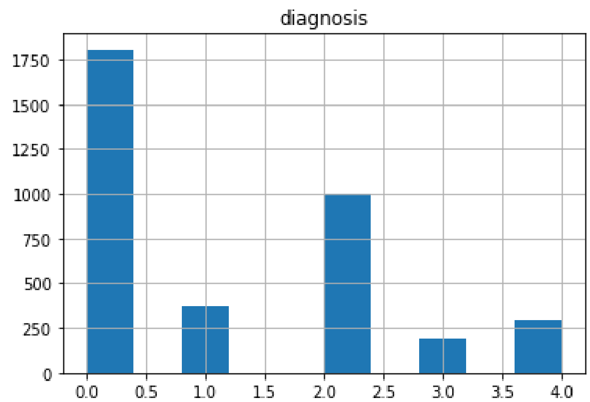


Chronically high blood sugar from diabetes is associated with damage to the tiny blood vessels in the retina, leading to diabetic retinopathy. The retina detects light and converts it to signals sent through the optic nerve to the brain. Diabetic retinopathy can cause blood vessels in the retina to leak fluid or haemorrhage (bleed), distorting vision. In its most advanced stage, new abnormal blood vessels proliferate (increase in number) on the surface of the retina, which can lead to scarring and cell loss in the retina.

Process:

Resnet101 is a deep learning model which had 101 Convolution layers that are good at identifying key features or patterns from the images given to the model. Resnets are the short form of Residual networks which are kind of Deep Neural Networks. Every block of the layer is a combination of Activation function called Relu and Batch norm layer for speeding up and to update the model weights.

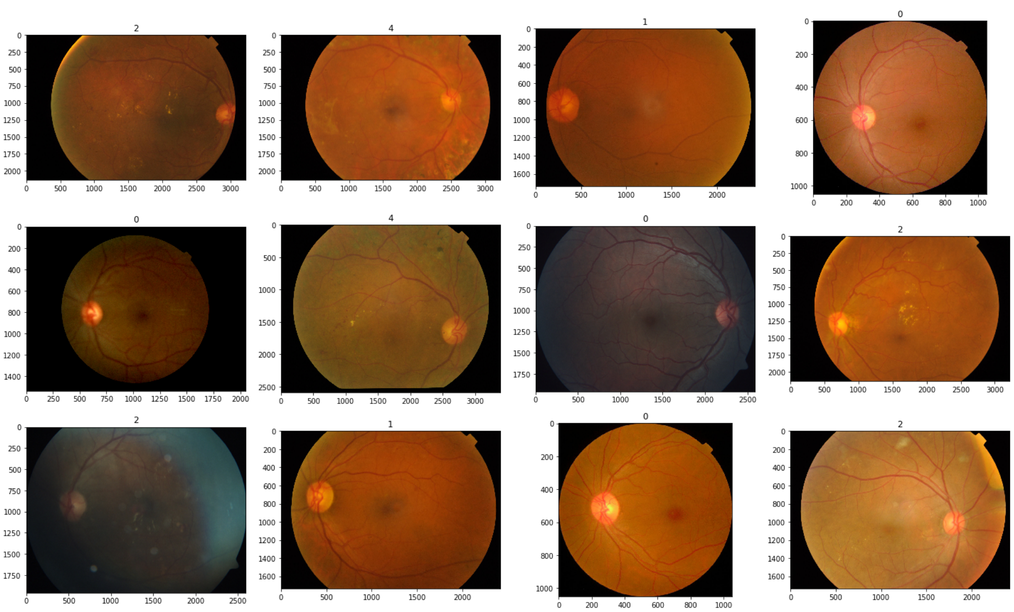
The train data we have is labelled and the distribution of train data is as shown below. The scale on the x-axis tells about the labels/stages that our images are and the scale on the y-axis tells about the number of images we have.



Out of 3500 total images of train data, we have 1800 images with No Diabetic retinopathy and we have 1000 images with moderate Diabetic retinopathy.

Sample Data:-

These samples show a collection of all 5 stages of images. We can see that all images here do not have the same radius from the centre. Few are not clear and dark, where others are clear and sharp.



As discussed above we use Resnet101 model with the input size of 256 x 256 in shape. We had several specifications that play an important role in model architecture and helps in training data such as

- Kernel size of 7x7 with 64 filters.
- Optimizer = ADAM
- Learning rate of optimizer = 0.001
- Batch size = 16
- Epoch size = 15
- Loss metric = Mean Square Error
- Activation layers
    - Relu
    - Max pooling

For every epoch, it runs through a set of 229 images which after 15 epochs sums up to 3500 images and at each epoch we have a batch size of 16 images pass through the model.

We used Adam optimizer that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based on training data.

As we have less amount of images such as 3500 to train the model, which in turn gives us less accuracy due to models inability to learn from less amount of data. Hence we performed Random horizontal flip, which makes our dataset diverse and helps the model to work with some unseen images of any rotation in test data. We then transformed the model weights to tensors and did standardisation.

We are testing our model with the test set of 1900 images with the batch size of 32, wherein each batch we have 61 images that sum up to approximately 1900 images. We used Adaptive average pooling that will reduce your data along the spatial dimension to the output/target size we want, which is 1 here that gives us outputs of classes like 0,1,2,3 and 4

You can see below table that gives the general view of model architecture with all its layers and kernel sizes. All the layers including input and output sum up to 101 which says about layers when we term resnet101.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Output:-

Sample output along with image id and respective predicted label by the model are below.

1	id_code	diagnosis
2	0005cfc8afb6	2
3	003f0afdcd15	3
4	006efc72b638	2
5	00836aaacf06	2
6	009245722fa4	3
7	009c019a7309	2
8	010d915e229a	2
9	0111b949947e	0
10	01499815e469	3

Conclusion:- Further the Accuracy of the model can be increased by adding more data to train, clean and clear data, Data augmentation, and Data transformation. In future, this work can also be extended in identifying Cataract and Glaucoma.