Diabetic Retinopathy

CLASSIFIYING THE DISEASE USING DEEP LEARNING MODELS

CLL 788 Term Paper Report

Course Instructor: Prof Hariprasad Kodamana

Aditi Singh

2017CH10188

Diabetic Retinopathy: Literature Review

Diabetes mellitus or more commonly know as ‘diabetes’ is a form of metabolic disease which happens due to high blood sugar levels in your body. It is a very common disease in India though it is highly dangerous disease, India has the 2nd largest population of diabetics followed by China. In many of the diabetes patients over time a certain eye condition known as Diabetic Retinopathy is developed which causes gradual vision loss over time. It affects blood vessels in retina where augmented blood vessels, fluid drip, exudates, hemorrhages, and micro aneurysms are observed. This can be detected via examining images of retina.

Since diabetes is a very common diseases among Indians, it becomes highly important that there should be regular checkup for diabetic retinopathy (DR). Right now, evaluation of images may remain a complex task, but advancing field of machine learning in computer vision can be utilized to diagnose the extent of infection. It can be used to train the model efficiently using deep learning frameworks and can use to detect the level of damage done to the eye by DR. These deep learning convolutional neural network models are able to quantify the blood vessels, fluid drip, exudates, hemorrhages, and micro aneurysms in the retinal image and hence can detect the severity of DR.

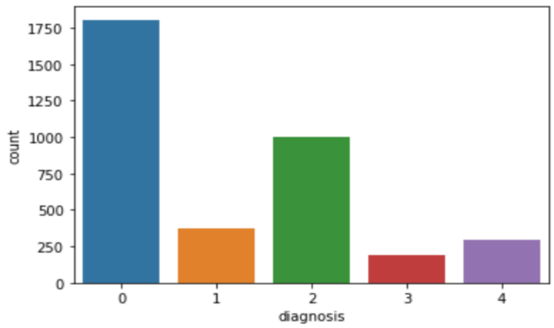
There are four stages through which Diabetic retinopathy usually progresses:

1. Mild non-proliferative retinopathy, occurrence of micro aneurysms (MA) and is the initial starting stage of DR.
2. Moderate non-proliferative retinopathy, where the blood vessels of the retina start losing their ability to transport the blood.
3. Severe non-proliferative retinopathy, deprived blood supply to retina due to the blockage of increasing number of blood vessels causing retina to secrete increasing number of fresh blood vessels.
4. Proliferative diabetic retinopathy (PDR), is the advanced stage, here the growth features secreted by the retina activate proliferation of the new blood vessels which are weak and would thus bleed and leak often the grow inside covering of retina in a vitreous gel, filling the eye. The associated scar tissue may contract further causing retinal detachment, leading to permanent loss in vision

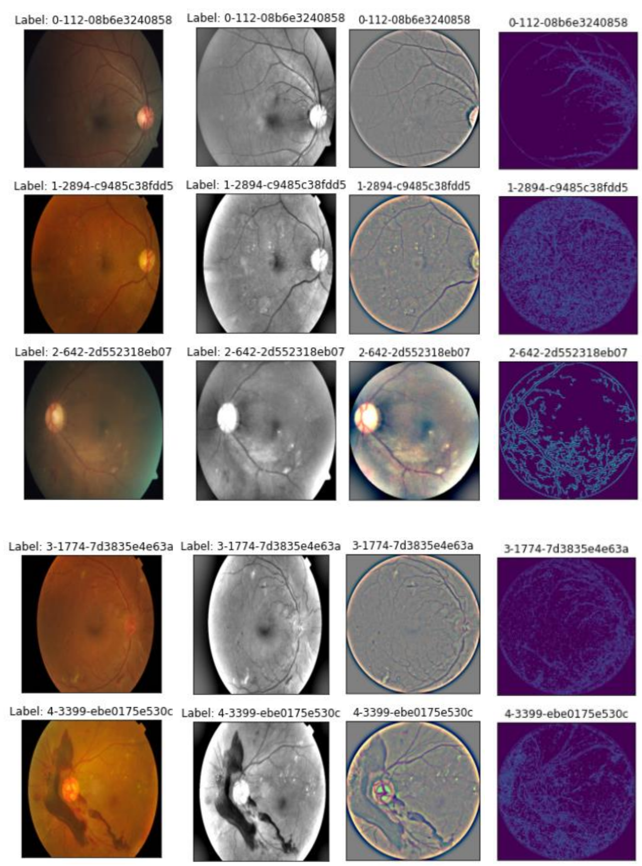
A lot of research has already been done in the field of Diabetic Retinopathy in both medicine sciences and as well as in machine learning the related work in these fields, show that researchers have proposed and implemented various machine learning methods, but the comparative study among these deep learning methods is still lacking for as far as Diabetic Retinopathy is concerned. *Suvajit et al* (2018) (reference paper used) have quite well explored the field and have used both deep learning and feed forward neural network model to detect the extent of DR in patients. They used a dataset comprising of 35000 train data images and 15000 test data images, for preprocessing the images they used edge detection and median filter methods for removing extra noise and making pictures easier to train by the model. Fuzzy C means was employed to detect the feature loss incurred after resizing the images. This term paper tries to make a deep learning model for diagnosing Diabetic Retinopathy using Python’s powerful artificial intelligence libraries such as Keras and TensorFlow. Next, I will be presenting the methodology of preprocessing the images which has been done using PIL, OpenCV and skimage libraries and using 3 models to train the dataset and compare the results and then finally comparing the results obtained with the one from reference paper.

Preprocessing the Dataset

Dataset used was downloaded from Kaggle with 3662 train images and 1928 test images. The image size was around 2136x3216x3 but I reduced them to 224x224 for avoiding memory issues for the RAM storage. The class labels are 5 which are as: 0 - No DR, 1 - Mild DR, 2 - Moderate DR, 3 - Severe DR, 4 -Proliferative DR, below is the frequency of these labels in the train dataset (0 is most frequent while 3 is least frequent)



In order to make the features of the retina more accurate, I gray scaled the train images followed by, cropping the images (so that more of the retina is visible) along with applying gaussian filter for each train image and then finally detecting the visible edges (edge detection)



1st column shows the Label wise (0 to 4) image of retina the 2nd column shows the same set of images with gray scaling and gaussian blur, the 3rd column shows the images being cropped and the 4th column shows edge detected images. (All the images are generated by the code)

Now after applying the above changes to every train image, next step was to apply the data augmentation to the train images using ImageDataGenerator library, there were 2 reasons to do so, first that the dataset was small and secondly the labels count were not the same hence it would help in improving the accuracy of the model. With this data preprocessing was done. (The code for image pre-processing is presented in 'termpaper_cll788_preprocessing.ipnyb')

Defining the Deep Learning Models

For training the pre processed images I used deep learning models. 3 models were used and the accuracy of the models were compared to check for the best model which correctly predicted the corresponding labels for each image. A powerful open-source software library, Keras was used to build these models. Keras provides a python interface for artificial neural networks it supports multiple backends such as TensorFlow (also used to build our models, Keras have acted as an interface to it), Microsoft Cognitive Toolkit, Theano, and PlaidML, all these libraries are again used for building up artificial neural networks such as multilayer NNs, CNNs

Model 1: Single channel 2-dimensional Sequential 2-layer convolutional neural network

The size of input array used in this model is (3662,224,224,1). Every convolutional layer has 3x3 kernel size and 32 units is for first layer and then there is 64 units in CNN for next layer. Max Pooling is done after layer to reduce the extent of features being lost along with padding, for every layer RELU activation is applied at the end there is dense layer where the output is computed and which has 5 neuron (corresponding to 5 class labels) , SOFTMAX activation function has been used. Also, dropout layer has also been added to add dropout to the input. The model is compiled using 'Categorical_Crossentropy' function for computing the loss, Adam optimizer is used and accuracy metrics is used.

Model 2: Three channel 2-dimensional Sequential 3-layer convolutional neural network

The size of input array used in this model is (3662,224,224,3). Every convolutional layer has 3x3 kernel size and 32 units is for first layer and then there are 64 units employed in the CNN for the next two layers. Max Pooling is done at every layer, for every layer RELU activation is applied at the end there is dense layer where the output is computed, SOFTMAX activation function has been used. The model is compiled using 'Categorical_Crossentropy' function for computing the loss, Adam optimizer is used and accuracy metrics is used.

Model 3: Pre-Trained MobileNetV2 Model (Transfer Learning)

This is a different kind of model from the previous two, it uses a pre-trained model on the ImageNet dataset, a large dataset consisting of 1.4M images and 1000 classes. This is known as transfer learning where one model which is trained for one assignment is again used and trained for the second for example in our case, this model is pre-trained for the ImageNet dataset and now it will be used for our task which is to classify the retina images into the severity of DR. The model is freezed (it is not updated by our training image data) and sequential model is made by adding dense and dropout layers.

Next step is fitting these models with our training data which has been split up between training (80%) and validation sets (20%), for the first 2 models, 10 epochs are used for iterations and 100 for the third model and then the accuracy and loss incurred by each of the set is computed and, on this basis, the three models will be compared. (The code for this is presented in 'termpaper_cll788_models.ipnyb')

Results and Discussion

Below are the plots of accuracy (training and testing) and loss vs epochs for each model.

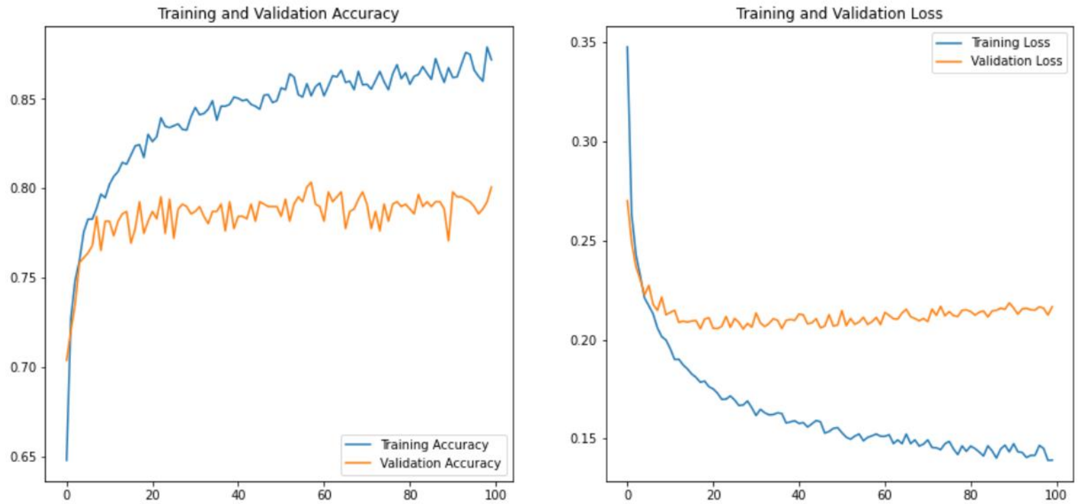
Model 1

Model 2

Following observations were made:

- Out of these two models, model 1 has slightly better training accuracy, while the losses for model 2 is lesser (0.21) this can be due to the fact that model 1 is simpler and thus it trains easily but model 2 since is using more channels and is more tightly built it prevents more features thus the losses are less.
- Validation accuracy for model 1 stays in the range of 0.7-0.72 while for model 2 it is a little higher for around 0.74 but it is also better since with each epoch, the accuracy improves for model 2 hence overall model 2 seems to be the better model for predicting the stage of diabetic retinopathy of the patient.

Model 3



- Comparing the results of model 3 with model 2 (the better of the previous 2 models) one can see that both the train set (0.88) and validation set (0.8) accuracy are very high for model 3 and with every epoch it also increases.
- Moreover, the training and validation loss is much lower for model 3 and it uniformly decreases with time unlike for the first two models.

Models Used	Epochs	Image Size	Activation Function	Learning Rate	Computation Time	Training Accuracy	Test Accuracy
Model 1	10	224x224	SOFTMAX	0.001	50 min	80.52%	71.21%
Model 2	10	224x224	SOFTMAX	0.001	1 hr	80.07%	74.62%
Model 3	100	224x224	SOFTMAX	0.001	1.5 hrs	87.59%	80.08%

Tabular Comparison of the 3 models used

Thus, out of the three models, Model 3 (pre-trained ImageNet model) is the most accurate model with high training and validation accuracy and low training and validation loss, though it takes the highest computation time out of the three but is the most effective one. Below are the accuracy percentages of the models employed in the reference paper.

Models Used	Training Accuracy	Testing Accuracy
BNN	62.7%	42%
DNN	89.6%	86.3%
CNN (VGGNET)	76.4%	78.3%

Tabular comparison for the accuracy for the 3 models in reference paper, Suvajit et al (2018)

- My best model (Model 3) has around 88% train accuracy and 80% test accuracy which is much similar to the ones (though test accuracy was lesser for my model) obtained with DNN model in the reference paper but there the models are trained for a much larger dataset compared to mine
- Due to less computation powers of my CPU higher computations and more complex data were not applied as compared to that mentioned in reference papers
- Also due to availability less number train images (1/10th of the dataset which was used in reference paper), my accuracy of the model was affected since a greater number of train images would definitely improve the model accuracy.

Conclusion

The term paper tries to propose a deep learning model for detecting Diabetic Retinopathy disease based on the reference paper as well as it presents a literature survey on the disease. Was able to preprocess the train (3662) and test images (1928) by gray scaling, gaussian blurring, image cropping followed by edge detection (as in the reference paper). Trained the preprocessed dataset using 3 models (simple NN, 3 layer CNN and pre-trained CNN model) and compared the training and testing accuracies for each model. It was figured out that the best model with the highest train and validation (testing) accuracy and lowest train and validation (testing) loss was Model 3, that is the pre-trained model on the ImageNet dataset this is also termed as transfer learning model. The results of the model's accuracies were compared with the ones in the reference paper and it was figured out that they were decently close. The shortcomings in my model were majorly due small train images size and less computation power of local CPU. Path forward, will definitely improve on the models will try to get the larger dataset and will also use better techniques to preprocess the data

References

Research Paper

Classification of Diabetic Retinopathy Images by Using Deep Learning Models, Suvajit Dutta, Bonthala CS Manideep, Syed Muzamil Basha, Ronnie D. Caytiles and N. Ch. S. N. Iyengar, International Journal of Grid and Distributed Computing Vol. 11, No. 1 (2018)

Code References

- https://www.analyticsvidhya.com/blog/2019/01/build-image-classification-model-10-minutes/
- https://www.analyticsvidhya.com/blog/2020/10/create-image-classification-model-python-keras/
- https://stackoverflow.com/questions/13538748/crop-black-edges-with-opencv
- https://codereview.stackexchange.com/questions/132914/crop-black-border-of-image-using-numpy/132934
- https://www.kaggle.com/taindow/pre-processing-train-and-test-images
- https://www.pluralsight.com/guides/building-features-from-image-data-in-python