Detection of Diabetic Retinopathy

Project Report | DS5500 | Fall 2019

Prasanna Challa, Nikhar Gaurav, Anish Narkar, Prathwish Shetty
Northeastern University, Boston, MA

1. Summary:

Millions of people suffer from diabetic retinopathy (DR), the leading cause of blindness among working-age adults. Early detection of disease can prevent blindness among people living in rural areas where medical screening is difficult to conduct. Currently, technicians travel to these rural areas to capture images and then rely on highly trained doctors to review the images and provide diagnosis. Our goal is to develop a machine learning model capable of diagnosing DR based solely on retinal scans in order to reduce the amount of diagnostic work that is placed on medical experts.

The problem we are trying to solve is a multilabel image classification task predicting the severity of diabetic retinopathy on a scale of 0 to 4, shown in Table 1.

Table 1: Description of Severity of Diabetic Retinopathy

Scale	Severity of Diabetic Retinopathy
0	None
1	Mild
2	Moderate
3	Severe
4	Proliferative

2. Data

Data sets from 2 Kaggle competitions 2015[1] and 2019[2] are merged to increase the sample size for training. The data from 2019 alone has very few samples corresponding to high severity levels of DR, which is the reason we had to combine the datasets. The labels from both the competitions have 5 labels. The final dataset was divided into train set with 4800 images and test set with 1200 images. Considering the subjective nature of the problem, each Image is rated on a scale of 0 to 4 based on the severity of diabetic retinopathy by expert doctors. Images may contain artefacts, be out of focus, underexposed, or overexposed. The images we regathered from multiple clinics using a variety of cameras over an extended period of time, which will introduce further variation. The distribution of classes in the final data set were shown in Figure 1 and samples images from each class are shown in Figure 2.

Fig 1: Distribution of classes

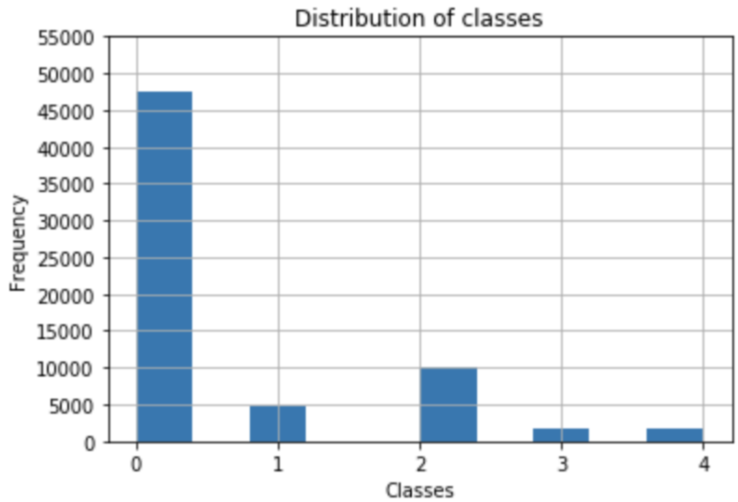
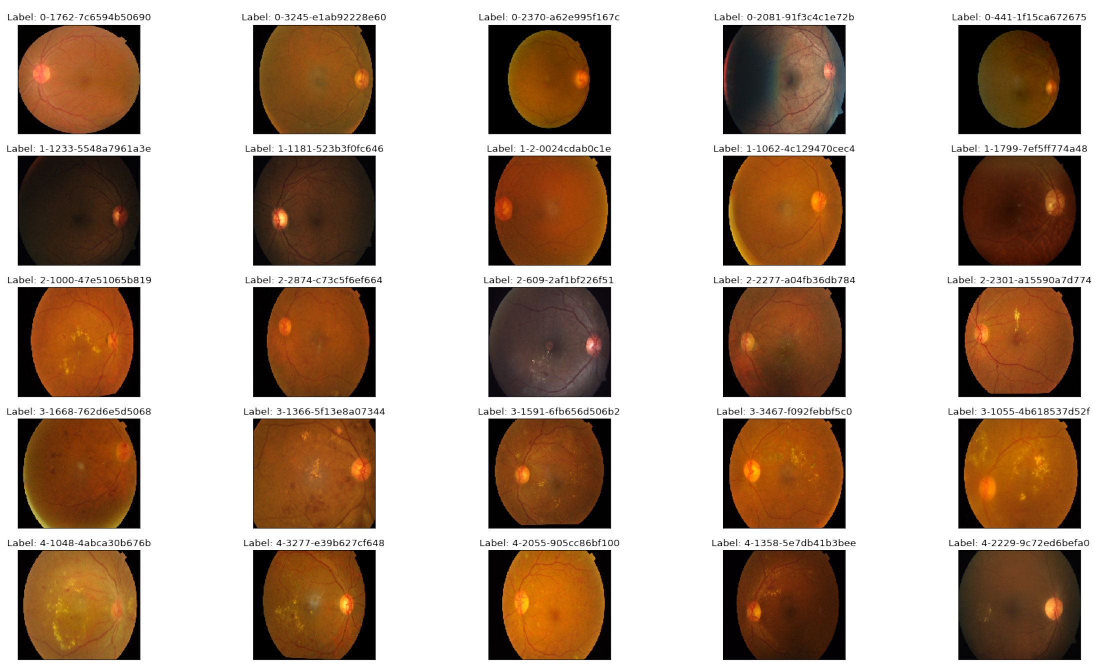
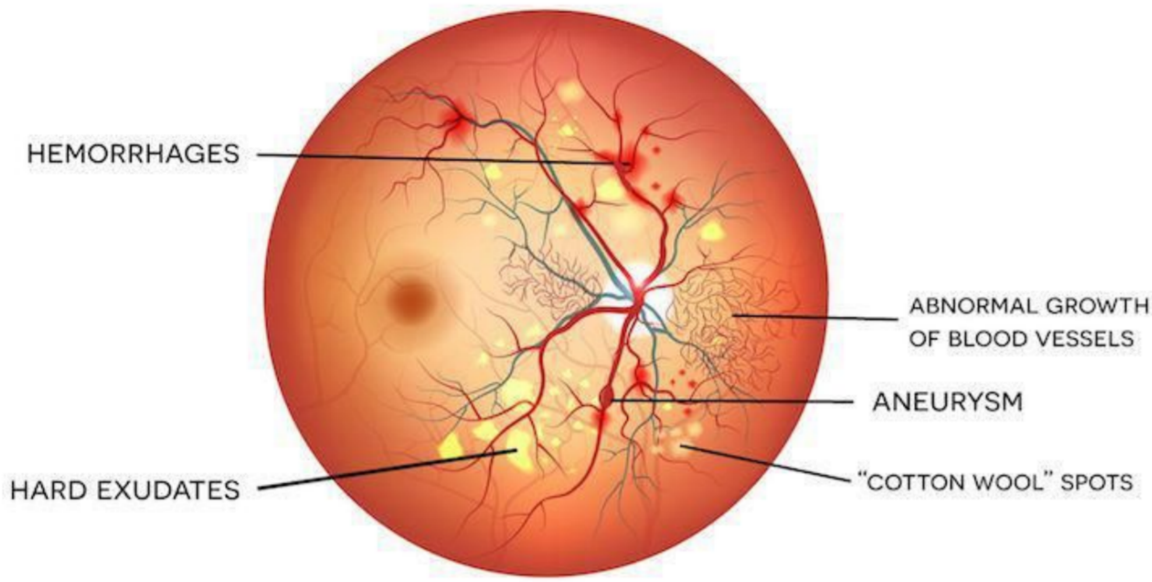


Fig 2: Retinal Scans for different severity of DR



There are 5 conditions that could help identify the disease, shown in Figure 3: Hemorrhages, Hard exudates, Abnormal growth of blood vessels, Aneurysm and Cotton wool spots. From visual inspection, we have found that the cases of Aneurysm and Abnormal growth of blood vessels are relatively hard to identify.

Fig 3: Factors causing DR



3. Methods:

3.1 Preprocessing:

Preprocessing the data involved various steps namely sampling, resizing, masking and weighted gaussian blurring. First, we sampled the images to handle class imbalance by selecting equal number of images from each class. Next we resize the images to the modelling requirements (i.e. 125x125, 256x256 or 512 x 512). To the resized images we added a circle mask to get rid of the improper mask used in few of the images. Finally, we used weighted gaussian blurring to accentuate the features of the images mainly the blood vessels.

3.2 Data Augmentation:

We augmented the i.e. make slight changes to every image at each epoch, this is done to improve network localization and reduce overfitting. We employ techniques like rolling, rotation, zoom and sheer to augment the data.

3.3 Convolution Neural Networks:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. Components of CNN:

1. Convolution Layer: The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. In context of CNN it is necessary to mention that each neuron only processes the data in its receptive field which is defined as the neurons in previous layers which are connected to the current neuron.
2. Max Pooling: Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Convolutional networks may include local or global pooling layers to streamline the underlying computation.
3. Fully Connected Layer: Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

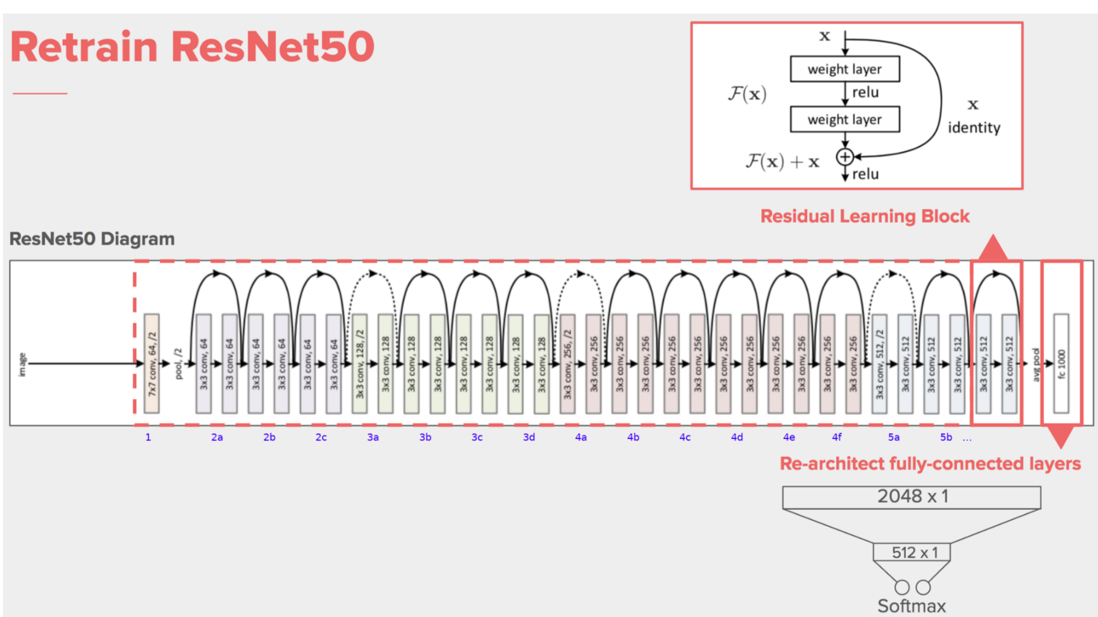
Based on the different configurations of these basic components, multiple architectures are possible. We have selected the following architectures:

3.3.1 Resnet50

When working with deep networks it is key to understand that simply increasing the depth won't result in improving the efficiency of the algorithm. On the contrary, it is possible that the performance of the algorithm would deteriorate with increase in depth (Factors such as Vanishing Gradient Problem contributes towards this issue). Resnet which stands for residual networks work around this problem with the use of "skip connections". Skip connections takes the activation of layer and passes it into deeper layers and not just the next layer. They allow the model to learn an identity function which ensures that the higher layer will perform at least as good as the lower layer, and not worse.

The ResNet-50 model consists of 5 stages each with a convolution and Identity block as shown in Figure 4. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.

Fig 4: Resnet Framework



3.3.2 XceptionNet

Xception was developed by Google as an improvement over the InceptionNet. While using CNN, there are several choices which are available to designers to optimize, this involves size of filters, strides for pooling and so on. InceptionNet rather than making a choice simply combines all the result of all these operations. These operations are referred to as depth wise convolution. In order to streamline this combined result InceptionNet uses 1x1 convolution which results in reduction in number of operations performed. This is referred as pointwise convolution. Thus, in InceptionNet the order of operations is depth-wise convolutions followed by pointwise convolution.

XceptionNet reverses the order of operations, with pointwise convolution followed by depth-wise convolution. This is claimed to be unimportant because when it is used in stacked setting, there are only small differences appeared at the beginning and at the end of all the chained inception modules. The other difference between these models. In the original Inception Module, there is non-linearity after first operation. In Xception, the modified depth-wise separable convolution, there is NO intermediate ReLU non-linearity. Apart from these, XceptionNet also leverages skip connections to deal with degradation problem with increase in depth of the network.

4. Evaluation metrics:

In order to evaluate the accuracy of our prediction we used three metrics. Quadratic Kappa Metric, Accuracy and F1 Score.

Quadratic Kappa Score: The Kappa coefficient is a chance-adjusted index of agreement. It quantifies the amount of agreement between an algorithm's predictions and some trusted labels of the same objects. Kappa starts with accuracy - the proportion of all objects that both the algorithm and the trusted labels assigned to the same category or class. However, it then attempts to adjust for the probability of the algorithm and trusted labels assigning items to the same category "by chance." It does this by assuming that the algorithm and the trusted labels each have a predetermined quota for the proportion of objects to assign to each category. The original kappa coefficient assumed nominal categories, but this was later extended to non-nominal categories through "weighting." The idea behind weighting is that some categories are more similar than others and thus some mismatching pairs of categories deserve varying degrees of "partial credit." Quadratic weights are one popular way of determining how much partial credit to assign to each mismatched pair of categories; there are other weights.

Accuracy: Number of correct predictions divided by total observations

F1 Score: Harmonic mean between Precision and Recall

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

5. Experimental Results:

The objective was revised to test the model performance on a relatively simpler problem of binary classification. Classes 0,1 were considered as having no DR and classes 2,3,4 as having DR.

We have tried using different optimizers including SGD, Adam, AdaGrad and AdaDelta. Best results were achieved using AdaDelta which could be due to better adaptive learning rate. We also used drop-out to reduce overfitting of train data and various loss functions to improve the predictive performance. The major factors that helped improve kappa score were increasing the resolution of images to 512*512, balanced training set and using pretrained weights of ImageNet. The loss function for XceptionNet which is our best performing model were shown in Figure 5 for binary classification and multiclass classification.

Comparison of models for binary and multiclass classifications are shown in Table 2.

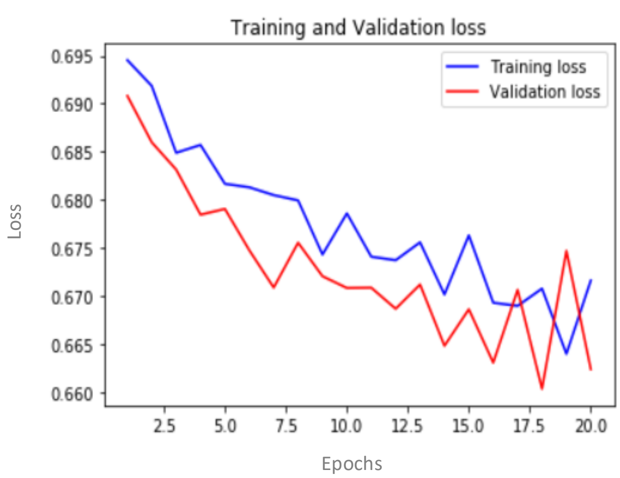
Table 2: Results

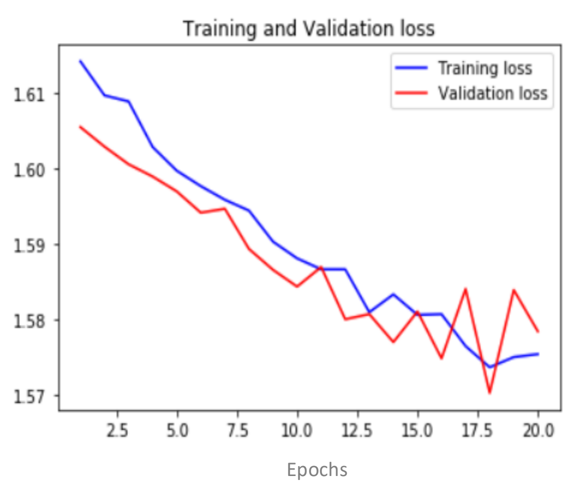
Classifier Type	Model	Data	Kappa	Accuracy	Precision	Recall
Binary	XceptionNet	Train	0.86	0.89	0.89	0.90
		Test	0.81	0.87	0.86	0.86
	ResNet50	Train	0.81	0.85	0.76	0.75
		Test	0.77	0.80	0.79	0.82
MultiClass	XceptionNet	Train	0.69	0.74	0.72	0.76
		Test	0.62	0.68	0.65	0.64
	ResNet50	Train	0.64	0.71	0.71	0.72
		Test	0.60	0.63	0.62	0.63

Best hyperparameter combination:

Optimizer - Ada delta (0.01), Size - 512*512, Dropout - 0.3, Loss function - Cross entropy

Fig 5: Binary and Multiclass loss functions in first and second plots respectively





6. Discussion and Conclusion

After completion of this project we made certain observations about our methodology and approach which could improve our model. Pre-processing data is a significant step and must be done carefully in order to generate efficient model. In case of image data performing image augmentation along with image pre-processing yields better results. Another important factor which affects the efficiency of models is the choice of optimizer during gradient descent. Since there are many such optimizers available with standard libraries, it is necessary to experiment with these as well as we discovered that certain optimizers gave us better results. Finally, choice of metric remains one of the most crucial steps while dealing with classification problems, this is especially true while dealing with imbalanced data. As far as possible improvements are concerned, we can perhaps experiment with different techniques for image preprocessing. Also considering the wide range of CNN framework which are available, we can always try new frameworks and create an ensemble model based on all the techniques, which is known to yield efficient results.

7. Statement of contributions:

- Prasanna Kumar Challa was responsible for working on XceptionNet, generate results and tune hyperparameters.
- Nikhar Gaurav was responsible for working on XceptionNet, generate results and tune hyperparameters.
- Anish Narkar was responsible for defining the methods and tune hyperparameters.
- Prathwish Shetty was responsible for data preprocessing and augmentation, work on ResNet50, tune hyperparameters and improve the results.

8. References:

[1] APTOS 2019 Blindness Detection.
[2] APTOS : Eye Preprocessing in Diabetic Retinopathy.
[3] François Chollet, Xception: Deep Learning with Depthwise Separable Convolutions,2016.
[4] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, Deep Residual Learning for Image Recognition,2015.
[5] Christian Szegedy and Wei Liu and Yangqing Jia and Pierre Sermanet and Scott Reed and Dragomir Anguelov and Dumitru Erhan and Vincent Vanhoucke and Andrew Rabinovich,Going Deeper with Convolutions,2014.