Automated Detection of Diabetic Retinopathy using Deep Learning

By Dheeraj Malepati: 1216633262; Vamsee Sundar Vipparthi: 1217319337; Madhu Priyatam PV: 1215125015

Introduction

In the modern era, there are lots of diseases that affect the normal life of a human. One such disease is Diabetes, which occurs due to the increasing insulin levels in a human body. It tends to affect multiple organs of the human body as it is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels. As per the statistics of WHO[2], In 2014, 8.5% of adults aged 18 years and older had diabetes. In 2016, diabetes was the direct cause of 1.6 million deaths and in 2012 high blood glucose was the cause of another 2.2 million deaths.
The different types of diabetes are

Type 1 diabetes:
Type 1 diabetes is characterized by deficient insulin production and requires daily administration of insulin. The cause of type 1 diabetes is not known and it is not preventable with current knowledge. Symptoms include excessive excretion of urine (polyuria), thirst (polydipsia), constant hunger, weight loss, vision changes, and fatigue.[2]

Type 2 diabetes:
Type 2 diabetes results from the body's ineffective use of insulin. Type 2 diabetes comprises the majority of people with diabetes around the world, and is largely the result of excess body weight and physical inactivity.Symptoms may be similar to those of type 1 diabetes, but are often less marked.[2]

Gestational diabetes:
Gestational diabetes develops in some women when they are pregnant. Most of the time, this type of diabetes goes away after the baby is born. However, if you've had gestational diabetes, you have a greater chance of developing type 2 diabetes later in life.[2]

Other types of diabetes:
Less common types include monogenic diabetes, which is an inherited form of diabetes, and cystic fibrosis-related diabetes

The most common consequences of diabetes are as follows:

- Adults with diabetes have a two- to three-fold increased risk of heart attacks and strokes.
- Diabetes can damage the heart, blood vessels, eyes, kidneys, and nerves.
- Combined with reduced blood flow, neuropathy (nerve damage) in the feet increases the chance of foot ulcers, infection and eventual need for limb amputation.
- **Diabetic retinopathy**[2] is an important cause of blindness, and occurs as a result of long-term accumulated damage to the small blood vessels in the retina. 2.6% of global blindness can be attributed to diabetes.
- Diabetes is among the leading causes of kidney failure.

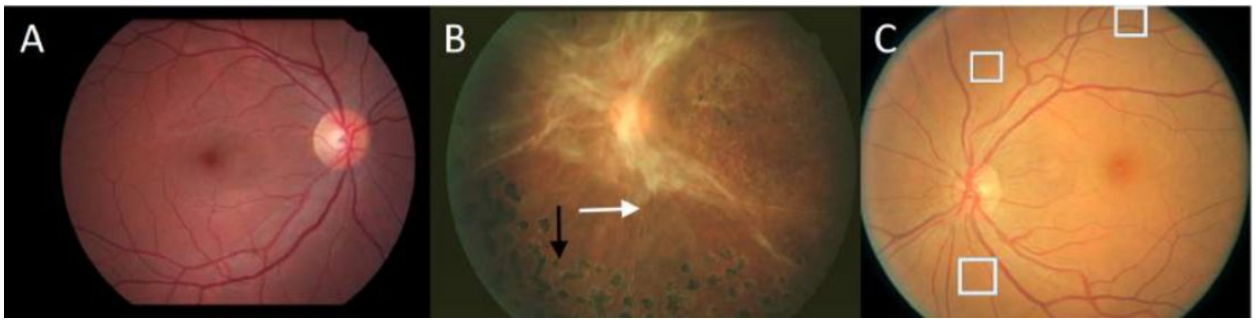
In this context, we will discuss about Diabetic Retinopathy. When diabetes affects human eyes, the disease is termed as diabetic retinopathy (DR). The blood vessels in a human are very small in size and hence more susceptible. The corrosion of blood vessels starts occurring when the blood sugar levels are increased much above the normal levels for a prolonged period of time. This pathological condition is referred to as DR .Patients suffering from this disease over a long period of time, get affected by a disease as diabetic retinopathy (DR) which attacks the retina .DR affects the blood vessels in the retina. It is a leading cause for various eye diseases and may even lead to blindness. There are three phases of diabetic retinopathy, these are: i) non-proliferative diabetic retinopathy (NPDR) (ii) proliferative diabetic retinopathy (PDR) (iii) maculopathy .The mildest phase of DR is NPDR, in this phase, small blood vessels within the retina leak fluid or blood. The mildest phase of DR is NPDR, in this phase, small blood vessels within the retina leak fluid or blood. The leaking fluid causes the retina to swell, developing small dot like deposits known as microaneurysms and haemorrhages .The next phase is PDR, in which new vessels grow on the inner surface of the retina leading to visual impairment.

Problem Description

A major percent of patients with diabetes who were recommended for eye screening actually obtained a screening exam. Manual methods of diabetic retinopathy are very time-consuming and require a trained clinician to examine and evaluate the digital color fundus photographs of the patient's retina. Since there is an immense global need for these screening programs, manual methods are unable to meet the demand. Computer-aided diagnosis of DR has been explored in the past to reduce the manual burden of the clinicians and opthamologists. Automated techniques of DR play a major role in solving these types of problems and all the past efforts of image classification, pattern recognition and machine learning have made excellent progress.
To understand more about the problem, below are few medical terminologies.

1. Fundus: Interior surface of the eye or interior lining of the eyeball including the retina.
2. Fundoscopy: An exam that uses a magnifying lens and a light to check the fundus of the eye.
3. Microaneurysms: Local disturbances in the diameter of the blood vessels of the retina. In visual format, they appear as small dots which are often in clusters form.

The diagnosis of Diabetic Retinopathy depends on the presence of microaneurysms, retinal hemorrhages(escape of blood) on the fundus of the eye.



Fig(1). Representative retinal images of DR at various stages of the disease, as labeled: A- normal, B- end stage, C- early stage. Arrows in B point to pathological indications. White boxes in C enclose very small lesions that the CNNs have difficulty discerning.

All the figures above, retinal images of DR at different stages are shown. Figure A shows the normal stage, figure B shows the end stage and figure C shows the early stage. Automated methods are active in the area of machine learning to detect microaneurysms and grade the

fundoscopic images. In the past, research on CNN studies including kNN and Support Vector Machines for large datasets showed a sensitivity of 90% for binary classification of normal or mild/moderate/severe. But accuracy for classifying into four classes of Diabetic Retinopathy differ and depend on the type of disease class. The four classes of Diabetic Retinopathy are:

1. R0: No signs of Diabetic Retinopathy
2. R1: Mild nonproliferative diabetic retinopathy (NPDR)
3. R2: Moderate NPDR
4. R3: Severe NPDR

R0 and R3 stages are capable of high sensitivity and R1 and R2 are low. We have implemented an automatic DR grading system which is capable of image classification based on disease pathologies from four severity levels and which determines the sensitivity and specificity of 4-ary classification models. To enhance sensitivities of mild and early classes, we are planning to train the multi-class models including various methods of data preprocessing and data augmentation. In this project, we will be training and testing two CNN architectures, AlexNet and GoogLeNet, and will be applying batch normalization, L2 regularization, dropout to perform optimally. Further, to increase the accuracy we apply transfer learning on CNN. We will be using two datasets from kaggle and Messidor-1 for training and testing purposes.The models will be implemented in Tensorflow using modified versions of open source code.

Methodology

1. Preprocessing: Pre-processing of an image is always required to extract the features in a more clear and proper manner. It is also required to remove any noise present in the image.

a. **Convert color images to grayscale to reduce computation complexity:** This is because in many objects, color isn't necessary to recognize and interpret an image. Grayscale can be good enough for recognizing certain objects. Because color images contain more information than black and white images, they can add unnecessary complexity and take up more space in memory. Color images are represented in three channels, which means that converting it to grayscale reduces the number of pixels that need to be processed.

b. **Image cropping using Otsu's method:**
Otsu's method is used to perform automatic image thresholding. In short, the algorithm returns a single intensity threshold that separates pixels into foreground and background. Otsu's algorithm tries to find a threshold value (t) which minimizes the weighted within-class variance given by the relation :

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

where

$$q_1(t) = \sum_{i=1}^{t} P(i) \quad \& \quad q_1(t) = \sum_{i=t+1}^{I} P(i)$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \quad \& \quad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \& \quad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_1(t)]^2 \frac{P(i)}{q_2(t)}$$

Algorithm:

1. Compute histogram and probabilities of each intensity level
2. Set up initial wi(0) and μi(0)
3. Step through all possible thresholds t = 1, 2, 3… maximum intensity
    i. Update ωi and μi
    ii. Compute σb2(t)

4. Desired threshold corresponds to the maximum σ b 2 (t)

c. **Image Normalization:**

In image pre-processing, normalization is a process where the range of pixel intensity values are changed. The purpose of normalization is to bring all the images into a range that is more familiar and achieve consistency.
There are three common techniques for value normalization:

1. (x - x.min()) / (x.max() - x.min()) # values from 0 to 1
2. 2*(x - x.min()) / (x.max() - x.min()) - 1 # values from -1 to 1
3. (x - x.mean()) / x.std() # values from ? to ?, but mean at 0

In our project, we normalized the images by subtracting the minimum pixel intensity from each channel and dividing by the mean pixel intensity to represent pixels in the range 0 to 1.

d. **Real Data augmentation:**

Data augmentation is a strategy that enables machine learning models to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks. In most approaches, basic types of augmentation are used in training neural networks. In our project, we augmented the number of images in real-time during the training phase to improve network localization capability and reduce overfitting. During each epoch, a random augmentation of images that preserve collinearity and distance ratios was performed. We implemented random padding with zeros, zoom, rolling and rotation. These transformations are particularly effective when applied to disease class R1 as it is the most difficult class to grade and very few in number.

e**. Contrast Adjustment**

We have used Adaptive histogram equalization (AHE) to improve contrast in images. Ordinary histogram equalization computes a global equalization whereas an adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is suitable for improving the local contrast and enhancing the definitions of edges in each region of an image. Contrast Limited AHE (CLAHE) differs from adaptive histogram equalization in its contrast limiting. In the case of CLAHE, the contrast limiting procedure is applied to each neighborhood from which a transformation function is derived. CLAHE was developed to prevent the over amplification of noise that adaptive histogram equalization can give rise to.

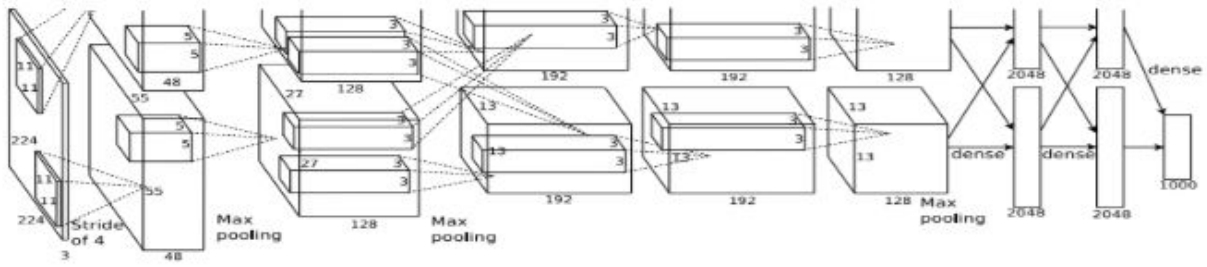
2. CNN Architectures:

A Convolution neural network(CNN) is a deep neural network, which employs the linear mathematical operation called ”Convolution” on the inputs.CNN is the most influential

innovations in the field of Computer Vision in recent times.The main advantage of a CNN is that the amount of the preprocessing that should be done on an image before feeding into the network is very less when compared to other Computer Vision methods.The network learns the filters that will define the features in an image. They were inspired by the functioning of Visual Cortex(the part of the brain responsible for vision). The whole idea was the result of an experiment by Lettvin et al.25 in 1959 where they showed that some individual neuron cells in the visual cortex area fired only if the edges of a certain orientation are present. For example, some neurons fired when exposed to vertical edges and some when shown horizontal or diagonal edges. Lettvin et al. found out that all of these neurons were organized in a columnar architecture and that together, they were able to produce visual perception. This idea of specialized components inside of a system having specific tasks (the neuronal cells in the visual cortex looking for specific characteristics) is one that machines use as well, and is the basis behind CNNs. There are many CNN architectures evolved in the recent past, especially object recognition and classification tasks.In this project we have explored three state of the art CNN architectures, namely GoogLeNet,AlexNet and VGG19.These three architectures are winners and runner up of ImageNet Large Scale Visual Recognition Challenge(ILSVRC) held by Stanford University every year for object recognition,classification and localization task.

AlexNet[5]:

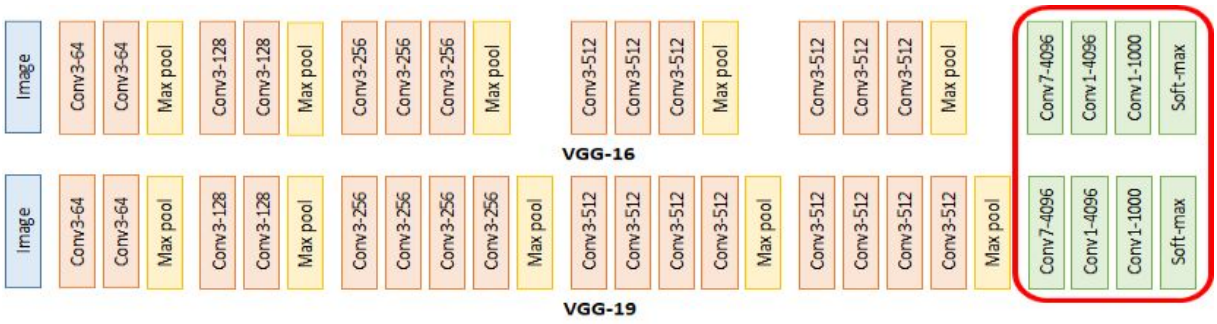
AlexNet is a winner of ILSVRC 2012, which achieved a top-5 error of 15.3%, more than 10.8 percentage points lower than that of the runner up.This network contains 7 layers , the first five are convolution layers some of them are followed by max-pooling and remaining two layers 60 million parameters. An important feature of AlexNet is the use of ReLU(Rectified Linear Unit) Nonlinearity. Tanh or sigmoid activation functions used to be the usual way to train a neural network model. AlexNet showed that using ReLU nonlinearity, deep CNNs could be trained much faster than using the saturating activation functions like tanh or sigmoid.



Fig(2.) Architecture of AlexNet

VGG-19[4]:

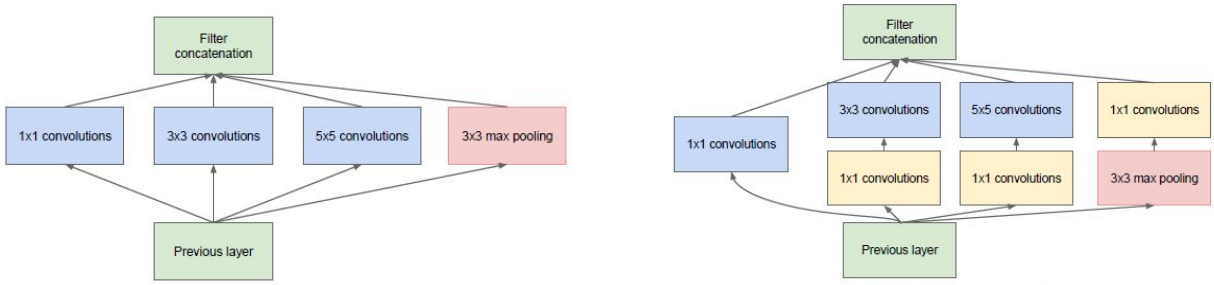
VGG-19 is the 1st Runner up of ILSVRC(ImageNet Large Scale Visual Recognition Competition) 2014 challenge.The network achieved a top-5 error of 10%.Developed by renowned Visual Geometry Group(hence the name) of Oxford University led by Andrew Zisserman.They have explored and proved that by increasing the depth of a CNN or ConvNet, the accuracy can be increased. The depth was increased by adding extra convolution layers and this is feasible because of the smaller filter size,i.e., 3x3.It contains 17 hidden layers and 144 million parameters.



Fig(3.) Different Layers in VGG

GoogLeNet[3]**:**

GoogLeNet is the winner of ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC2014).It is developed by Google in collaboration with UNC Chapel Hill, University of Michigan and Magic Leap Inc. The network achieved a top-5 error of 6%. This architecture consists of 22 layers deep. It reduces the number of parameters from 60 million (AlexNet) to 4 million, which is almost 12 times fewer parameters than Alexnet at the sametime more accurate although the network is bigger. At the heart of the Googlenet is the Inception module.The idea of the inception layer is to cover a bigger area, but also keep a fine resolution for small information on the images. So the idea is to convolve in parallel different sizes from the most accurate detailing (1x1) to a bigger one (5x5). We used 1x1 convolutions for dimensionality reduction before expensive convolutions.9 Inception modules are also stacked up.



Fig(4.) Overview of Inception Module

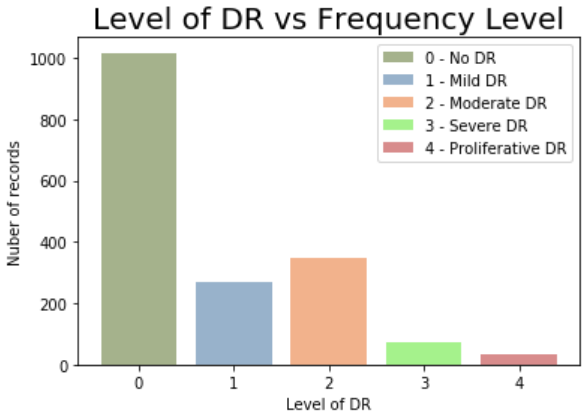
3. Datasets:

Physician verified messidor dataset:

We have used two fundoscope datasets for training the classifier in this project. **Messidor dataset:** MESSIDOR stands for **M**ethods to **E**valuate **S**egmentation and **I**ndexing **T**echniques in the field of **R**etinal **O**phthalmology. The 1200 eye fundus color numerical images of the Messidor database were acquired using a color video camera mounted with a 45 degree field of view. Images were captured using 8 bits per color plane at 1440*960, 2240*1488 or 2304*1536 pixels. 800 images were acquired with pupil dilation (one drop of Tropicamide at 0.5%) and 400 without dilation. The 1200 images are packaged in 3 sets, one per ophthalmologic department. Each set is divided into 4 zipped subsets containing each 100 images in TIFF format and an Excel file with medical diagnoses for each image.

Class Labels:

- 0 (Normal): (μA = 0) AND (H = 0)
- 1: (0 < μA <= 5) AND (H = 0)
- 2: ((5 < μA < 15) OR (0 < H < 5)) AND (NV = 0)
- 3: (μA >= 15) OR (H >=5) OR (NV = 1)
  μA: number of microaneurysms, H: number of hemorrhages, NV = 1: neovascularization
  NV = 0: no neovascularization



Fig(5.) Bar graph showing the distribution of Messidor dataset

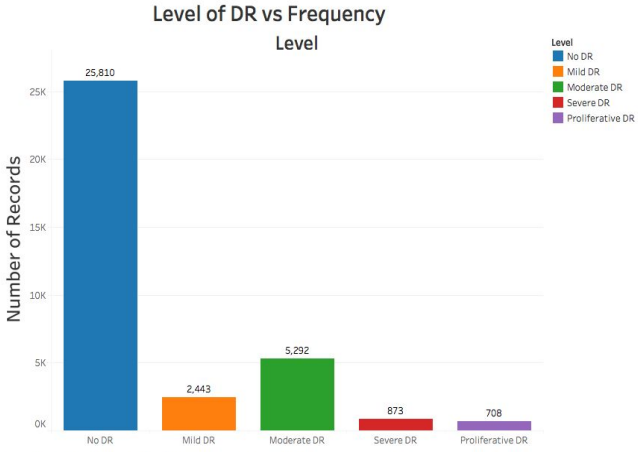
The class distribution of the Messidor dataset is shown in the figure above.As there are very few images for classes R2 and R3, real-time data augmentation is used to increase the number of fewer available images.

Publicly available Kaggle dataset:

This dataset has a large set of high-resolution retina images taken under a variety of imaging conditions. A left and right field is provided for every subject. Images are labeled with a subject id as well as either left or right.A clinician has rated the presence of diabetic retinopathy in each image on a scale of 0 to 4, according to the following scale:

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

The class distribution of the Kaggle dataset is given in the figure below:



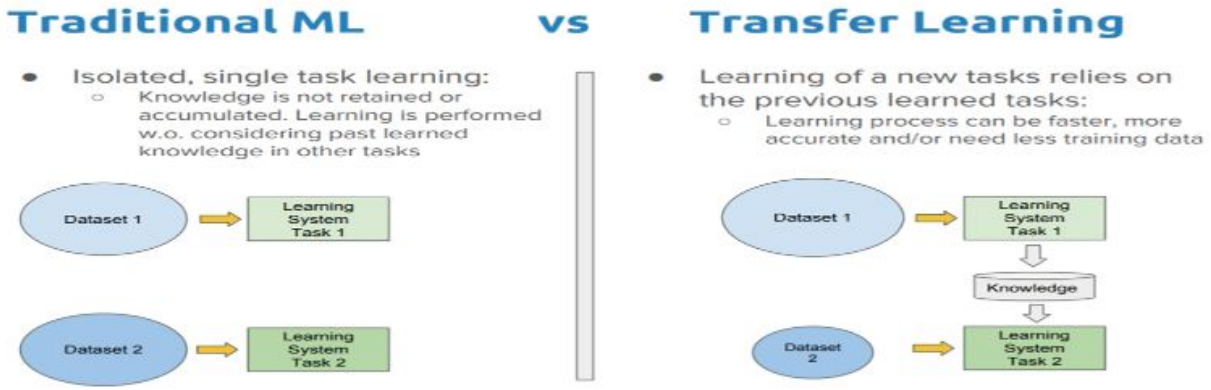
Fig(6.) Bar graph showing the distribution of Kaggle dataset

4. Training and Testing the Model:

We trained and tested the state of art CNN architecture models on Messidor and Kaggle dataset. We performed our experiments on Google Colab which provided us with a Tesla K80 GPU hardware device. Our results were designed in such a way that advanced visualization like bar charts and line charts and confusion matrix could be generated.

5. Transfer Learning:

A common misconception in the deep learning community is that without huge amount of data, it is not possible to create effective deep learning models. While data is a critical part of creating the network, the idea of transfer learning has helped to lessen the data demands. Transfer learning is the process of taking a pre-trained model (the weights and parameters of a network that has been trained on a large dataset) and fine-tuning the model with your small dataset[10]. The idea is that the pre-trained model will act as a feature extractor. In general, the last layer of the network is replaced with your classification layer (depending on how many classes in your problem). Then train (adapt) the network normally.

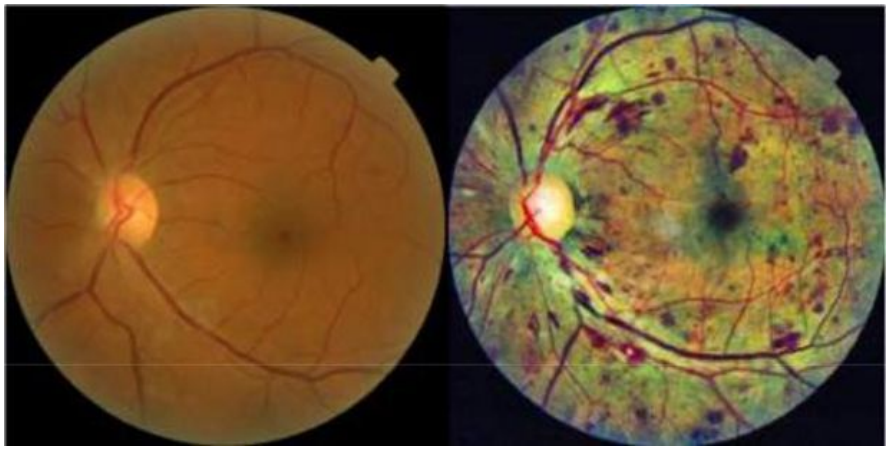


Fig(7.) Difference between Tradition ML and Transfer Learning

Results

1. Data Pre-processing:

The main step in the preprocessing is the Contrast Limited Adaptive Histogram Equalization(CLAHE) that enhances the contrast and supports in detecting the subtle features which are helpful in detecting the mild DR.The result of applying CLAHE on one of the retinal images is shown below.



Fig(8.) Retinal image before and after preprocessing

a. Binary Classification

For the binary classification task, AlexNet, VGG19 and GoogLeNet were trained. The binary-labelled data comprised of two classes. Class 1 has the retinal images of Moderate DR or Severe DR or Proliferate DR. The training for binary classification was performed on tensorflow in Google Colab platform.Class 2 has the retinal images of No Diabetic Retinopathy or Mild Diabetic Retinopathy. There are four parameters that are useful to evaluate the model performance on a test data set. Let us define **no disease(Normal or mild)** as negative class(class 1) and **moderate to end** stage of the disease as positive class(class 2). A **true positive** is an outcome where the model *correctly* predicts the *positive* class. Similarly, a **true negative** is an outcome where the model *correctly* predicts the *negative* class.A **false positive** is an outcome where the model *incorrectly* predicts the *positive* class. And a **false negative** is an outcome where the model *incorrectly* predicts the *negative* class.

TP = number of true positives
TN = number of true negatives
FP = number of false positives
FN = number of false negatives

These are often depicted using confusion matrix: Each column contain the number of predicted test samples for each class and Each row contains the number of samples that actually belong to the corresponding class.The format of confusion matrix is as follows:

	Class 1 Predicted	*Class 2 Predicted*
Class 1 Actual	**TP**	**FN**
Class 2 Actual	**FP**	**TN**

Fig(9.) Table depicting the TP,TN,FN,FP

Furthermore,in Medicine, Sensitivity, Accuracy and Specificity are the often used metrics to evaluate a model . These are defined as follows using the parameters that are defined before: **Sensitivity** is the ability of a test to correctly classify an individual as ′diseased′. The formula is as follows:

Sensitivity = $\frac{TP}{TP+FN}$ = Probability of being test positive when disease present.

Accuracy is the fraction of predictions that the model got correct of all the test samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Specificity is the ability of a test to correctly classify an individual as ′not diseased′. The formula is as follows:

$$\text{Specificity} = \frac{TN}{TN + FP} = \text{Probability of being test negative when disease is not present.}$$

The tables below depict the confusion matrix for AlexNet, GoogleNet on Messidor and Kaggle dataset.

Predicted Calss / True Class	**Class 1**	**Class 2**
Class 1	169	14
Class 2	26	115

Table:1 Confusion Matrix for ALexNet on Messidor Dataset

Predicted Calss / True Class	**Class 1**	**Class 2**
Class 1	164	19
Class 2	12	129

Table:2 Confusion Matrix for GoogLeNet on Messidor Dataset

Predicted Calss / True Class	**Class 1**	**Class 2**
Class 1	3791	266
Class 2	324	2571

Table:3 Confusion Matrix for AlexNet on Kaggle Dataset

Predicted Calss / True Class	**Class 1**	**Class 2**
True Class 1	3904	153
True Class 2	244	2651

Table:4 Confusion Matrix for GoogLeNet on Kaggle Dataset

Table 5 depicts the sensitivity, accuracy and specificity of the datasets

	Messidor Data Set		Kaggle Dataset	
	AlexNet	GoogLeNet	AlexNet	GoogLeNet
Accuracy	81.84%	89.87%	91.52%	94.3%
Sensitivity	92.3%	89.61%	93.44%	96.22%
Specificity	81.5%	91.48%	88.80%	91.52%

Table:5 sensitivity, accuracy and specificity

2. Multi-label Classification:

In the Multi Classification task, we have evaluated the best out of the three architectures in the previous stage. GoogLeNet and AlexNet performed better when compared to VGG-19. So for this task, we trained only these two architectures on Kaggle dataset. The confusion matrix for the same is given below:

	**R0**	**R1**	**R2 or R3**
R0	1490	10	20
R1	210	20	70
R2 or R3	10	150	2020

Table:6 Confusion Matrix for GoogLeNet on Kaggle Dataset

	**R0**	**R1**	**R2 or R3**
R0	1372	66	82
R1	212	12	76
R2 or R3	25	159	1996

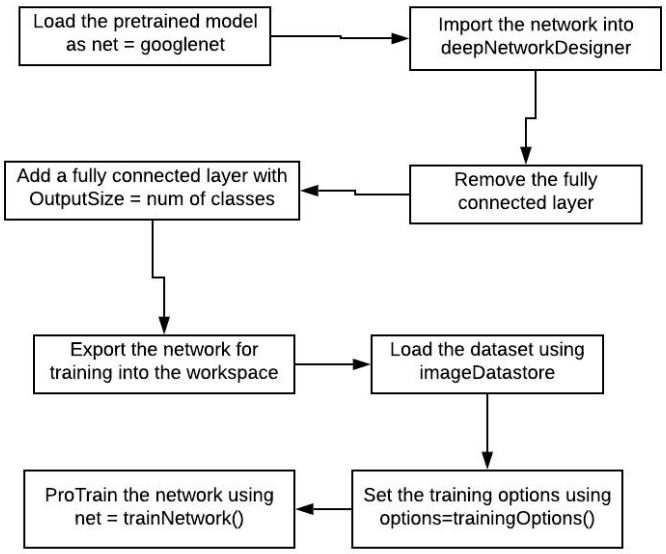
Table:7 Confusion Matrix for AlexNet on Kaggle Dataset

The accuracy for GoogLeNet is 88.5% and for AlexNet is 84.5%.As you can see the sensitivities of No DR and DR are 98% and 97%, but sensitivity of mild DR(R1) is 7%.This is due to the undetectable nature of the subtle features that contribute to mild DR and also we found out that there are inconsistencies with the labelling of Kaggle dataset.Thus we have to consider Messidor dataset which addresses this infidelity in the data.But the problem with the Messidor dataset is that the number of images are very less when compared to Kaggle dataset.

So to overcome this , we have trained the GoogLenet on Messidor dataset using transfer learning for 3-ary and 4-ary classification tasks.

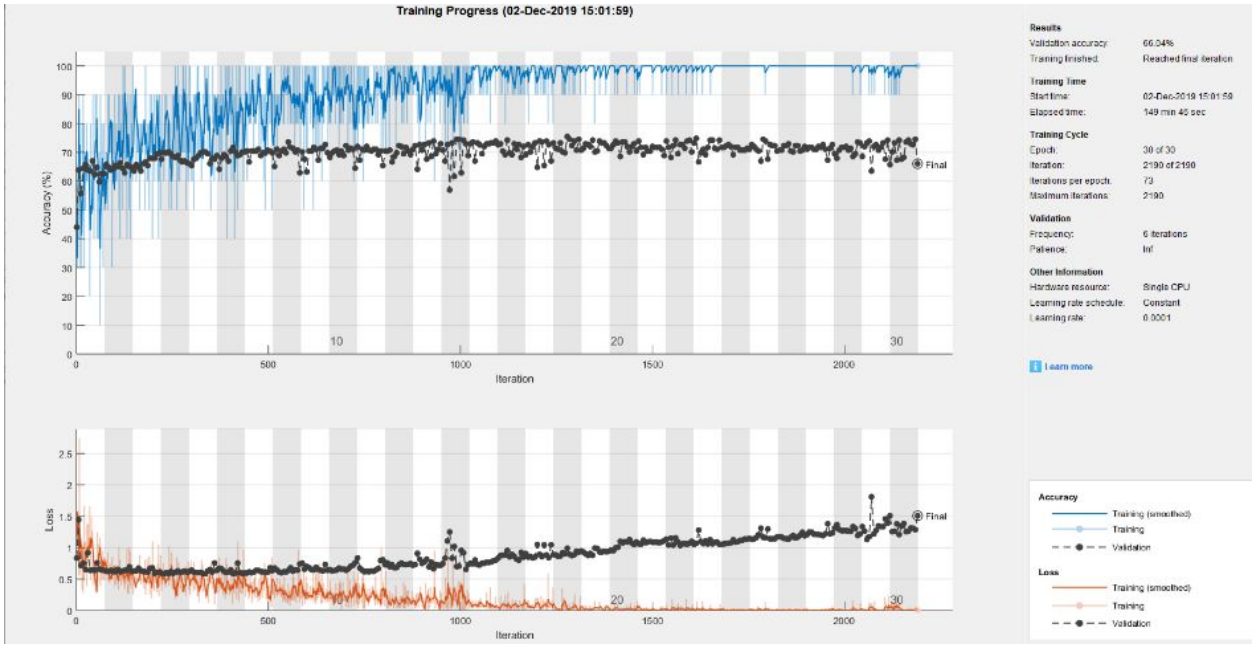
3. Transfer Learning

As mentioned in the previous section, we have used transfer learning to train GoogLeNet on Messidor dataset. To do the transfer learning we need the pretrained model, but since we couldn`t get the pretrained model in Tensorflow, we have used MATLAB`s pre-trained GoogLeNet model to train.This was trained on IMAGENET database for 1000 classes.As we do not need these many classes,we have to modify the architecture by replacing last fully connected layer with a new fully connected layer with the number of neurons equal to the number of classes for 3-ary and 4-ary classification problem.This was done using deepNetworkDesigner app in the DeepLearning Toolbox. The following are the steps that are followed to to the transfer learning:



Fig(10.) ProcessFlow of Transfer learning in MATLAB

The model was trained on the Messidor dataset for 30 epochs using stochastic gradient descent optimization with step decay learning rate initialized at 0.002. The classification model validation achieved 66.04% as the best accuracy.



Fig(11.) Training progress with Accuracy vs iteration and Loss vs iteration graphs

As you can see, the training led to an overfit of the model, to overcome this we have introduced L2 regularization,Dropout normalization and batch normalization.This also resulted in greater degree of convergence.And after which the test accuracy has jumped to 71%.

Conclusions and Future work:

All though we achieve state of the art accuracies for binary classification tasks, the accuracy and sensitivity decreases with an increase in the number of classes. One may think that increasing the dataset size is the solution to this problem, but the previous literature have suggested that in case of retinal images increasing the dataset size may not guarantee the increase in accuracy.

Visualizing the features that contribute to the mild and moderate DR suggests that the features are very subtle to be able to detected by the human eye. Even though we were able to successfully classify these into mild DR and moderate DR, but the sensitivities are not as high as the other classes. One suggestion is that the DR grading can be a two stage, where in the first stage the a computer vision is used to detect the regions that are responsible for various stages of DR and then pass through our model. This two stage process may increase the sensitivities of mild and moderate DR to some extent. Our future work is to explore the transfer learning method in depth to increase the detection of mild and moderate DR and improve the grading system.

References:

[1] Overview of Diabetes:
https://www.who.int/news-room/fact-sheets/detail/diabetes
[2] Overview of Diabetic Retinopathy:
https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611
[3] Overview of GoogLeNet:
https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7
[4] Overview of VGG-19:

https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11
[5] Overview of AlexNet:
https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637
[6] Overview of Transfer learning with Deep Network Designer MATLAB:
https://www.mathworks.com/help/deeplearning/ug/transfer-learning-with-deep-network-designer.html
[7] Datasets:
Kaggle : https://www.kaggle.com/c/diabetic-retinopathy-detection/data
Messidor: http://www.adcis.net/en/third-party/messidor2/
[7] M. D. Abr`amoff, J. M. Reinhardt, S. R. Russell, J. C. Folk, V. B. Mahajan, M. Niemeijer, and G. Quellec. Automated early detection of diabetic retinopathy. Ophthalmology, 117(6):1147–1154, 2010.
[8] B. Antal and A. Hajdu. An ensemble-based system for microaneurysm detection and diabetic retinopathy grading. IEEE transactions on biomedical engineering, 59(6):1720–1726, 2012.
[9] R. Gargeya and T. Leng. Automated identification of diabetic retinopathy using deep learning. Elsevier, 2017.
[10] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1717–1724, 2014.
[11] Y. Xu, T. Xiao, J. Zhang, K. Yang, and Z. Zhang. Scale-invariant convolutional neural networks. arXiv preprint arXiv:1411.6369,
2014.
[12] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579, 2015.
[13] Pratt H, Coenen F, Broadbent DM, Harding SP, Zheng Y. Convolutional neural networks for diabetic retinopathy. Procedia Computer Science. 2016 Jan 1;90:200-5.
[14] Ghosh R, Ghosh K, Maitra S. Automatic detection and classification of diabetic retinopathy stages using CNN. In2017 4th International Conference on Signal Processing and Integrated Networks (SPIN) 2017 Feb 2 (pp. 550-554). IEEE.