American University of Armenia

Zaven & Sonia Akian College of Science and Engineering

On Detecting Diabetic Retinopathy using Neural Networks and Machine Learning Techniques

Team:

Diana Sargsyan: diana_sargsyan@edu.aua.am

Lianna Puluzyan: lianna_puluzyan@edu.aua.am

Meri Abgaryan: meri_abgaryan2@edu.aua.am

Supervisor:

Michael Poghosyan, Ph.D.: mpoghosyan@aua.am

Spring, 2022

Abstract

Diabetic retinopathy (DR) is a disease that affects diabetic patients. The leading cause of the disease is the unmonitored high blood sugar levels. In the latter stages of the disease, patients might experience vision loss. Additionally,as some of our close relatives were diagnosed with DR, it is of huge importance to be able to detect the disease in early stage to prevent later complications. This paper discusses machine learning and neural network approaches of detecting the presence of the disease using fundus images of the eye. Algorithms includes SVM, KNN, Random Forests, and Convolutional Neural Networks.

Keywords: Machine Learning, neural networks, diabetic retinopathy, retina, fundus, SVM, KNN, Random Forest, Convolutional Neural Network.

Contents

Chapter 1

Introduction

1.1 Diabetic Retinopathy: Causes, symptoms and prevention

Diabetic retinopahty is a complication of diabetes affecting the retina of the eye [Cli00]. The primary cause of the disease is the damage caused to the blood vessels of the retina of the eye. At first, diabetic retinopathy has no prior symptoms, however parallel to the further progression of the disease, according to [Cli00] the symptoms of the disease include

- Spots or dark string in the vision

- Fluctuating vision

- Blurred vision

- Dark or empty vision

- Vision loss

Diabetic retinopathy occurs among patients struggling from type 1 or type 2 diabetes.

The vision loss among such patients is mostly a direct consequence of unmonitored high levels of sugar in the blood. However, development of diabetes during pregnancy increases the risk of developing diabetic retinopathy.

As stated in [Gro04], among 10.2 million US adults above the age of 40, the risk of having diabetic retinopathy is 40.3%, and the risk of vision loss as a severe consequence of diabetic retinopathy is 8.2%. On the other hand, approximately 4.1 million US citizens over the age of 40 are struggling from some stage of diabetic

retinopathy, and 1 in 12 have developed vision loss retinopathy. In the above mentioned study [Gro04], it is estimated that diabetic retinopathy will advance into a major public health problem in the future due to the cases of age specific diabetic retinopathy and ageing US population.

Prevention of diabetic retinopathy can be achieved by first of all, early detection of the disease.

1.2   Understanding Diabetic Retinopathy

Generally, diabetic retinopathy is divided into two subtypes, **early diabetic retinopathy**, or more commonly referred **nonproliferative**, in which case new blood vessels are not proliferating.
The other subtype is **advanced diabetic retionpathy**. In this case damaged blood vessels are closing of, thus causing the growth of novel abnormal blood vessels. Eventually, from the newly grown blood vessel, the scar tissue causes the retina of the eye to detach from the back of the eye. In addition to this, the novel abnormal blood vessels can interfere with the normal eye fluid, causing a damage to **optic nerve**, the latter results in **glaucoma** (vission loss).
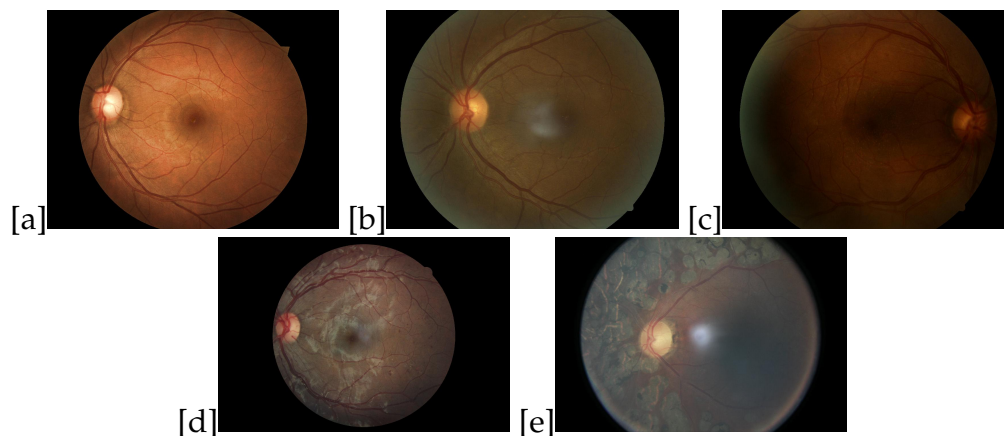

Figure 1.1: (a) Healthy eye (b) Mild Diabetic Retinopathy (c) Moderate Diabetic Retinopathy (d) Severe Diabetic Retinopathy (e) Proliferative Diabetic Retinopathy

1.3   Problem Setting and Project Motivation

Our project will discuss the problem of detection of diabetic retinopahy using machine learning techniques and neural network.

As mentioned above, diabetic retinopathy is a serious problem, with the intention of becoming a huge public health issue. Moreover, as some of our relatives experienced this disease, it motivated us to help in the detection and prevention of serious circumstance caused by the disease, including vision loss. Tackling the disease will improve the life quality of the patients. Moreover, we are hopeful that our project will assist doctor's in making more accurate diagnosis.

1.4   Related Work

In "Diagnosis of Diabetic Retionpathy Using Machine Learning Techniques", [PA13], Priya and Aruna use **PNN**(Probabilistic Neural Network), **Bayesian Classification** and **Support Vector Machine** to detect diabetic retionpathy. The detection is done based on features, such as **blood vessels,hemorrhages and exudates**. They propose a sequence of preprocessing steps, namely, **Gray Scale Conversion, Histogram Equalization, Discrete Wavelet Transform, Fuzzy C-Means Clustering and Gaussian Matched Filter Response** for blood vessel extraction. For hemorrhages and exudates **green channel extraction, thresholding and morphological operation-dilation** is used. Their approach exhibits accuracy of 97.6% for SVM.

V. Ramya proposed SVM based method for detection of diabetic retinopathy, [Ram18]. Ramya applies Histogram Equalization, afterwards training the dataset based on SVM. The accuracy of the model is 86%.

Sundhara Raja and Vasuki introduced an automatic detection for blood vessel to assist the detection of diabetic retinopathy. Their approach is based on the detection of **optic disc**[1]. Thus, as authors concluded in their paper, the segmentation of optic disc is crucial in extraction of blood vessel in detection of diabetic retinopathy. Authors perform the segmentation using **Anisotropic Diffusion Filter**, which is a non linear filtering techinque. The proposed method is tested on two datasets, **DRIVE** and **STARE**. Accuracy on the former dataset is 98.08%, compared to that of the former. 95.94%.

P.Sureka, B.Pavithra, et.al, [P.S19], presented a new approach on detection of the disease using **FP Growth algorithm**[2] and **PNN**. After applying preprocessing steps, similar to those of [PA13],they trained PNN and FP growth algorithm with accuracies 87.69% and 95.38% respectively.

---

[1]The centre of the optic disc is the source of origination of the blood vessels
[2]Frequency Pattern Growth algorithm is a pattern detecting algorithm that constructs an FP-tree, and then by divide and conquer obtains frequent item set [Sci20]

1.5   The Structure of the Paper

The rest of this paper is organized as follows: in the next chapter we present the data collection and preprocessing techniques.In chapter 3 we discuss the algorithms we have used on detection of the disease. In chapter 4, we analyze and measure the obtained results. In chapter 5, we conclude our findings and state the future work towards better results.

Chapter 2

Data and Preprocessing

2.1 Dataset

The dataset was obtained from Kaggle, [Neo21]. There are 34,408 pictures of the fundus of the retina in the dataset of JPEG format.The images are RGB, having 3 channels. It is divided into 5 classes, namely 0,1,2,3,4.

Table 2.1: The number of images of fundus of retina in respective classes

The class number	Severity of the disease	Number of fundus images
0	No DR	25800
1	Mild DR	2443
2	Moderate DR	5292
3	Severe DR	873
4	Proliferative	708

As we can see from the table above, the data is imbalanced, in class 0, there are 25800 images, that is approximately 5 times bigger, than the maximum sized folder of the remaining classes. Thus, in our algorithm we consider only approximately $\frac{1}{5}$ of the class 0, 4664 pictures. Next, we visualize our data using barplot.
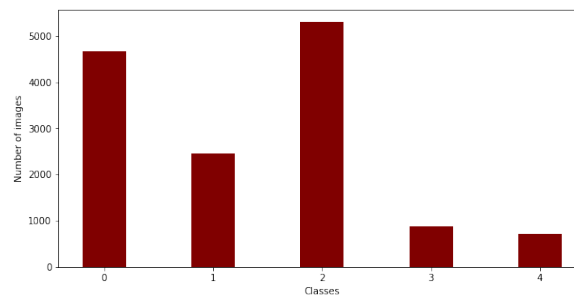

Figure 2.1: Barplot of distribution of fundus images among classes

2.2 Data Preprocessing: Attempt 1.0

After having a rough idea of the distribution of images, and making it slightly balanced, with proceed with the next step, that is preprocessing of the images. Similar to [PA13], we have used gray scale conversion and wavelet transform, however, afterwards we proceed with different preprocessing steps, namely

- Contrast Limited Adaptive Histogram Equalization

- Gaussian Blur

- Binary Thresholding

- Otsu's Thresholding

Now, let us explain what each of the preprocessing steps is and how it affects our images.

Gray scale conversion assigns a value of each pixel to be the amount of light. In our case, the conversion of RGB images into grayscale is done according to the following formula [Dyn19]

$$0.299 \cdot Red + 0.587 \cdot Green + 0.114 \cdot Blue$$

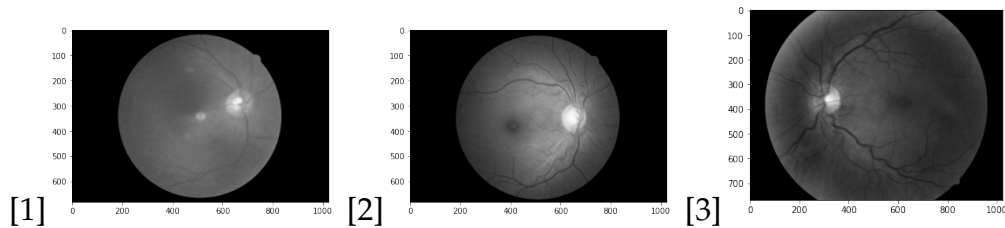After applying grayscale conversion, our images look like this

Figure 2.2: The result of applying grayscale conversion

The next step of preprocessing is applying Equalized Histogram. Histogram equalization is a simple, yet powerful technique. It works as follows, first constructs the histogram of image pixel intensities, and then evenly distributes the most frequent pixel values. Histogram equalization outputs images having higher contrast.However, Contrast Limited Adaptive Histogram Equalization (furthermore CLAHE) results in a higher quality contrasted images. This techniques divides the input image into $M \cdot N$ grid and then applies equalization to each cell
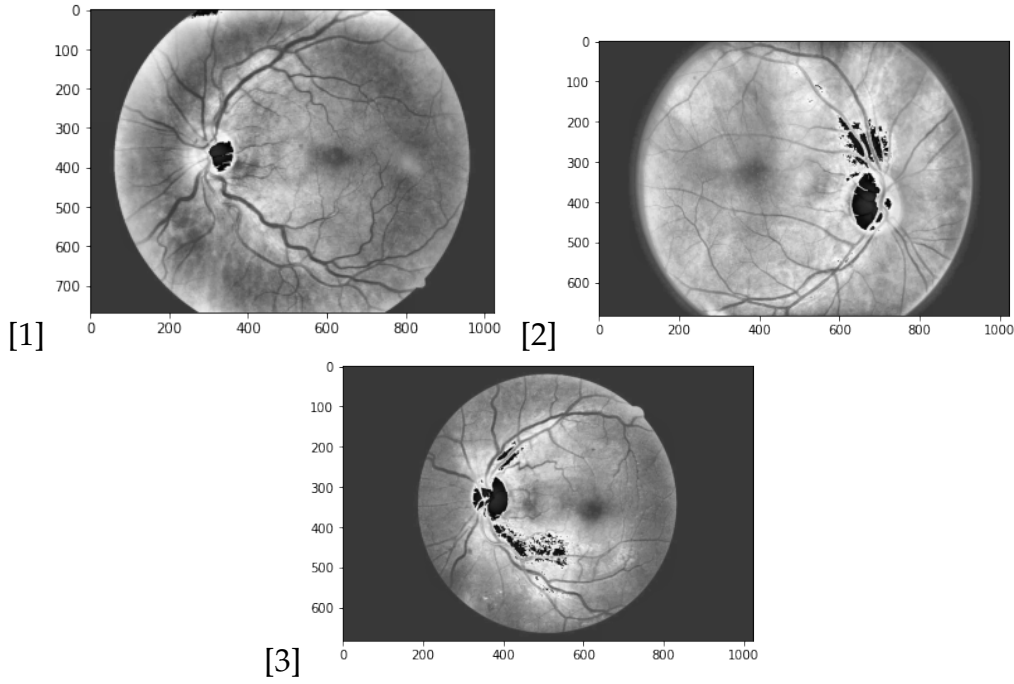
Figure 2.3: The result of applying CLAHE on gray scale images

in a grid [Ros21]. The result of applying CLAHE on our grayscaled images is the following

After obtaining CLAHE, we then proceed with Discrete Wavelet Transform. According to Mehdi Hosseinzadeh, [Hos20], a discrete wavelet transform (*DWT*) is a type of transform which decomposes a given signal into a number of sets, and each set represents time series of coefficients[1], describing the time evolution of the signal in the corresponding frequency band. Basis functions are called wavelets and they are obtained from a prototype wavelet $\psi(t)$, called mother wavelet by shifting. DWT2 is a Single-level discrete 2-D wavelet transform [PA13]. The input of the function is the eye image after CLAHE. DWT2 approximation coefficient matrix, horizontal, vertical and diagonal coefficient matrix. In our case, we have used Haar wavelets. The result after applying Haar DWR is as follows The result of Haar Transform is the reduction of the image size two times without the loss of crucial information.

The next step in the preprocessing, is Gaussian blur (Gaussian smoothing), that basically blures the image by a Gaussian function [LSA17]. It is a function of the form

---

[1]Time series data is collected for a single observation over time. There are several models and the notion of coefficients comes from that models
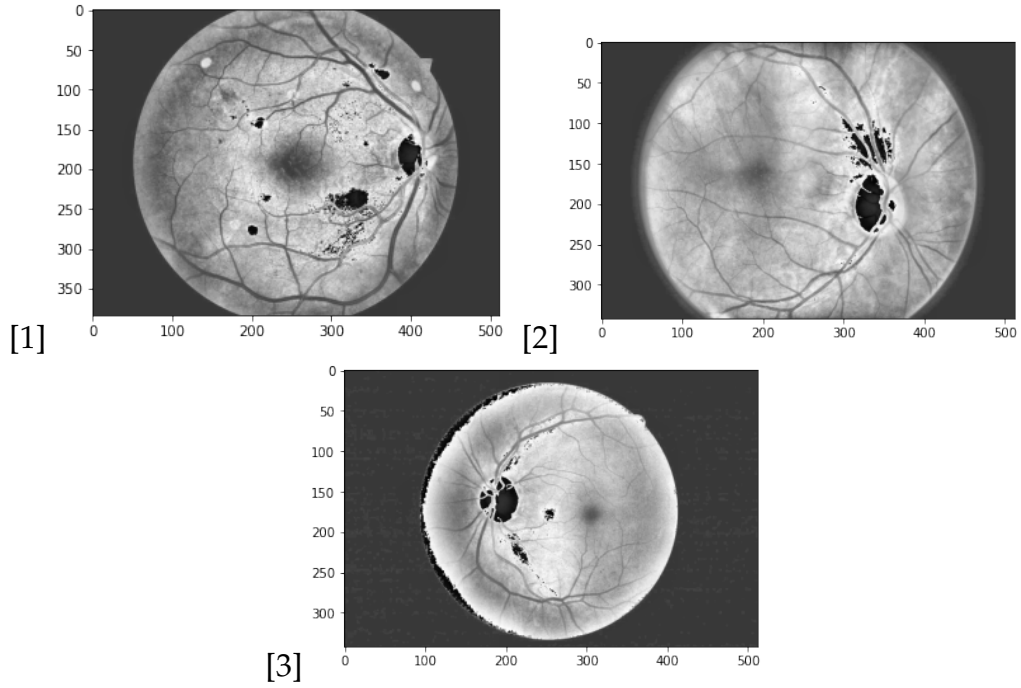
[1]　[2]

[3]

Figure 2.4: The result of applying Haar DWT on CLAHED images

$$f(x) = \alpha \cdot e^{-\frac{(x-b)^2}{2 \cdot c^2}}$$

where a,b and c are arbitrary real constants, s.t c is non-zero.

We use Gaussian blur is used to reduce the noise. Thus, after applying it on Haar transformed images look like this,
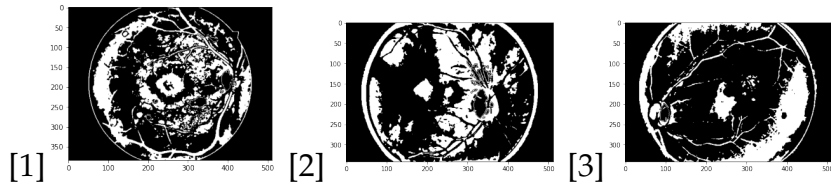

[1]　[2]　[3]

Figure 2.5: The result of applying Gaussian blur on DWT images

For the last two steps of preprocessing, we first use Adaptive Thresholding, then we apply Otu's Thresholding. Conventional thresholding works as follows, if the value of the pixel is less than the threshold, the value is set to zero for that pixel. We use adaptive threshold,since our image has different lightings in different areas, in which case the threshold for a pixel is determined according to the small region around the pixel [Vis]. The result of this algorithm on our blurred image is this,
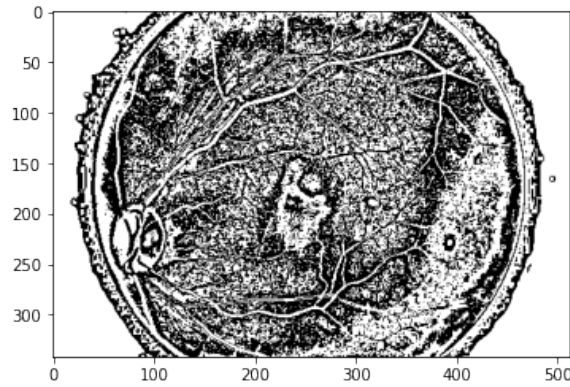
Figure 2.6: The result of applying adaptive thresholding on the blurred image

And finally, at the last step, we apply Otsu's thresholding,that in comparision to the traditional thresholding algorithms, that obtain the histogram of pixel values, Otsu's algorithm processes image histogram and then segments the objects by minimizing the variance for each of the classes [A.M20]. The main idea is dividing the histogram of an image into two separate clusters by using a threshold, that is the result of minimizing weighted variance of the classes. Otsu's thresholding was proposed in 1979, however up to nowadays it remains one of the important techinques in medical image segmentation.More detailed information on this technique can be found in Otsu's paper [Ots79]. The final preprocessed images are as follows
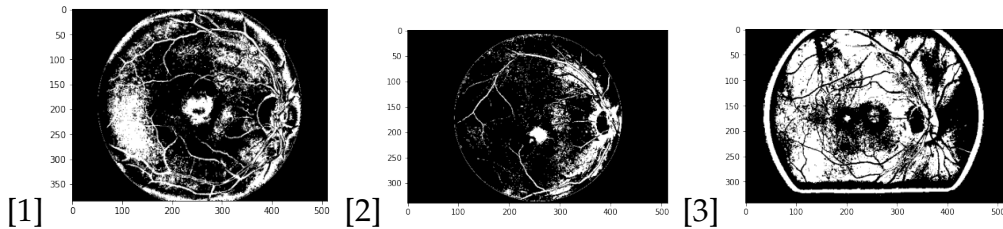

Figure 2.7: The final result after applying Otsu's thresholding

As we can notice from the final result, blood vessel, as well as exudates, (small white areas) are highly contrasted.However, as we shall discuss in the latter sections, with these preprocessing steps, the results are not very promising.

2.3   Data Preprocessing: Attempt 1.1

The preprocessing steps discussed in this section are quite similar to the first few steps in the former section. Specifically, images are converted to grayscale, CLAHE is applied, afterwards on the resulting images Haar discrete wavelet trans-

form is applied. In contrast to the previous preprocessing, here we decided to focus on the blood vessel extraction. Thus, as suggested in [PA13], we have used Gaussian Matched Filter(GMF). Since blood vessels have curvature-like shape, it is proposed that, Gaussian curvature can be used to detect vessel like structures at different orientations(note that, the definition of Gaussian curve was given in 2.2.). The Gaussian template is formed based on tuning of four different parameters. The main parameter is σ, a continuous value representing the spread of the intensity profile. The parameter **L** is the length of the vessel segment that needs to be processed, **T** is the position where the trails of the Gaussian curve will cut, and k represents the number of orientations, [ICA18]. The implementation of the GMF that was used in our code can be found here, [MAN17].

After applying GMF, we have used K-means clustering. This partitions our space into k different clusters. Thus, performing K-means will help us detect blood vessels, As a result of these preprocessing steps, the final input image looks like this,
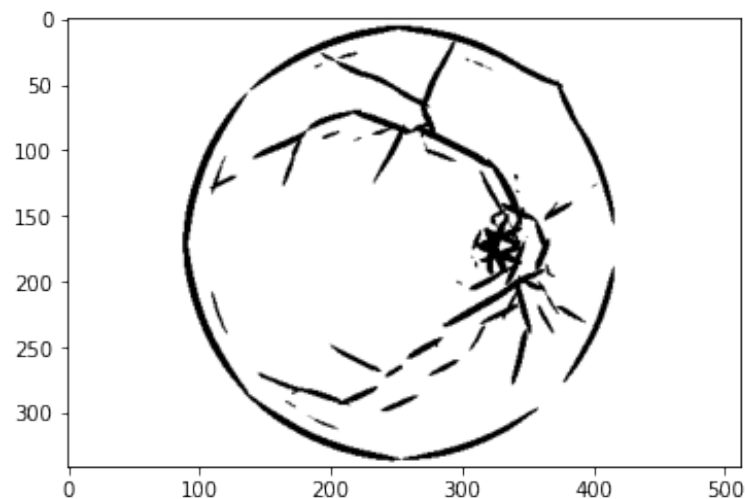

Figure 2.8: The final input image after GMF and K-means

Chapter 3

Algorithms and Models

3.1   Algorithms and Models

Our problem is in general a Classification problem, namely multi-class classification. Hence, we decided to implement several classification models. Further we will discuss all the algorithms in details.

At first, we have tried Convolutional Neural Network, namely Alex Net. Note that, in this case the input images given to the neural network are RGB of size 300X300. The loss is MSE, optimizer is SGD. However, the maximum accuracy we achieved was 30%. The main reason of such result is data imbalance.

Moreover, we tried also resizing images to 64X64, with **cross entropy loss** and optimization AdamW. In this case the obtained accuracy was 34%, which is quite low.

Afterwards, on our unprocessed RGB images we have tried Resnet-34, as a loss function we used cross entropy loss, and as optimizer, SGD. However, as mentioned above, the data was imabalanced, thus, in order to overcome this, we assign a weight to each class, depending on the quantity of the images contained in that class. The assigned weight is then multiplied with loss to balance the classes.

As for the classical Machine Learning models, on our preprocessed images (using all the preprocessing techniques discussed in 2.1 and 2.3) we have applied

- SVM

- KNN

- Random Forests

- Resnet-34(with balanced data)

Chapter 4

Results

4.1 Experiments and Results

4.1.1 Resnet-34

After receiving low accuracy on Alex-net architecture, we have decided to use Resnet-34, a pre-trained model, that was trained on Imagenet dataset. Resnet-34 has 34 convolutional layers. As opposed to traditional networks, Resnet-34 takes residuals from each layer and uses them in the subsequent connected layer. Here is the visualization of Resnet-34 layers, pretrained on Imagenet data set.
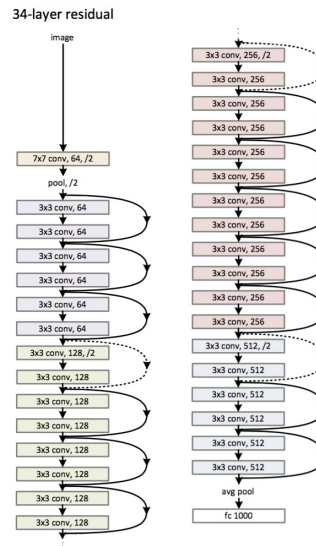

Figure 4.1: Resnet-34 architecture pretrained on Imagenet

As our problem is multi-class classification with 5 classes, we added 4 new layers, so that the final output corresponds to the mentioned 5 classes.
The images were unprocessed images with three channels, RGB, however we resized it to 64X64, normalized by subtracting the mean of dataset, and divide by

standard deviation. The loss function is MSE, optimizer is SGD. To overcome the class imbalance, we used class weights in the loss function. After training our data set the obtained accuracy is 85%. Bellow is the train-validation loss against the number of epochs.
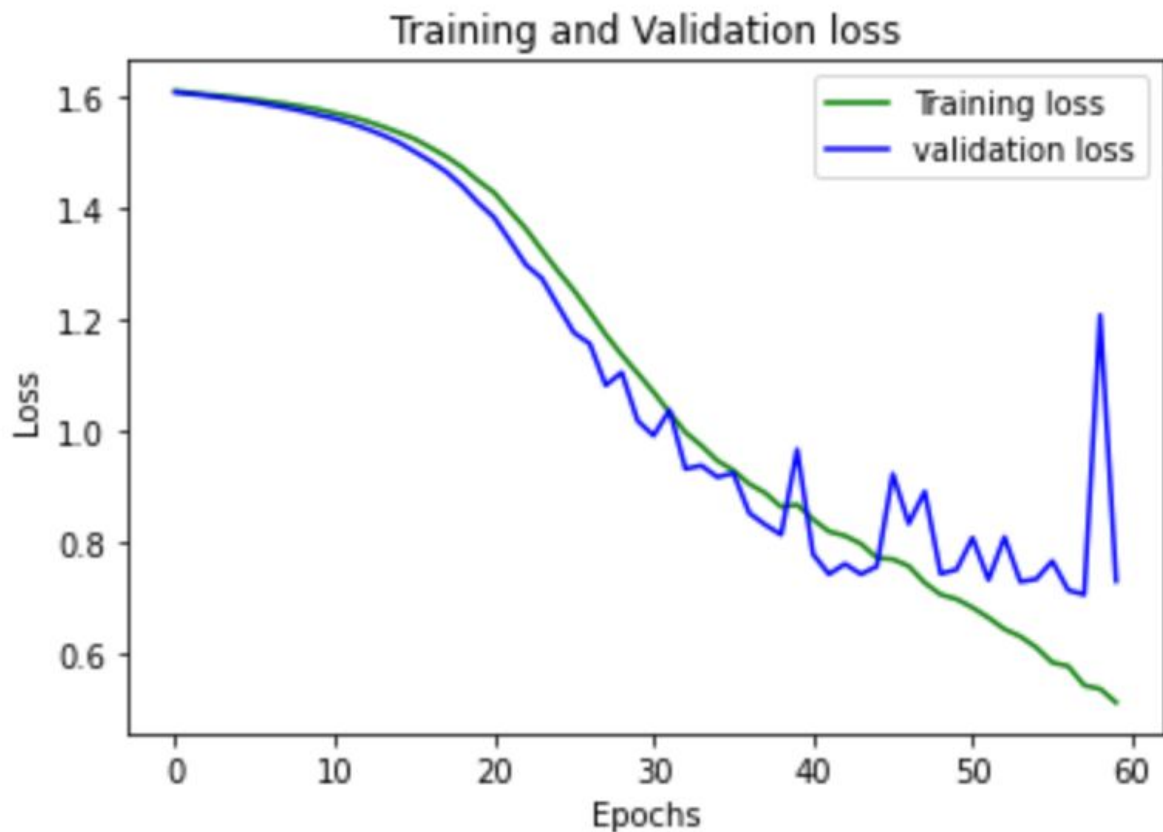

Figure 4.2: Train-validation loss against the number of epochs

4.1.2 SVM on preprocessed images

As stated in section 2.1, we have attempted two prepossessing techniques. As we have multiclass classification, we have used One-vs-rest Classifier(OvR), with SVC kernel RBF and linear kernel. To overcome the data imbalance, we have taken equal amounts of images from each class. Unfortunately, both of the preprocessing techniques proved to be ineffective. The accuracy score obtained by the preprocessing attempt 1 was only 23% with linear kernel, same as with RBF kernel. As a next step, we used bagging with SVC, obtaining 25% accuracy with linear kernel, with RBF 18%

After obtaining low accuracy scores, we proceeded with second attempt of

preprocessing, and applied the same models with same parameters as above. The following table represents the models, preprocessing technique along with respective accuracies.

Table 4.1: The obtained accuracies with respect to models, different parameters and preprocessing attempts, 2.2, 2.3

Model	Kernel	Preprocessing attempt	Accuracy
SVC	Linear	Attempt 1	23%
SVC	Linear	Attempt 2	21%
SVC	RBF	Attempt 1	23%
SVC	RBF	Attempt 2	23%
SVC with Bagging	Polynomial	Attempt 1	25%
SVC with Bagging	Polynomial	Attempt 2	15%

4.1.3 Random forests on preprocessed images

The next classification method is Random Forests with minimum samples leaves set to 5. Similar to SVM, we trained our model on two preprocessing attempts. Unfortunately, both of them performed poorly, accuracy is 26% for attempt 1, and 21% on the attempt 2. Since, random forests performed a little bit better compared to SVC, to visualize the number of misclassified images, bellow is the confussion matrix.
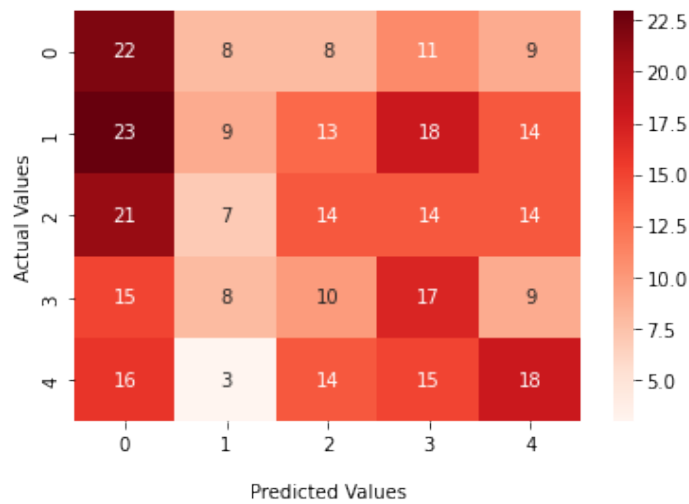

Figure 4.3: Confusion matrix of Random Forests trained of preprocessed images attempt 1

4.1.4   We don't give up: KNN

As our last hope, we trained our preprocessed images (both first and second attempts) on KNN, where $NN \in [1, 14]$. The highest accuracy for preprocessing attempts 1 and 2 is 26% and 24% respectively. Again for comparison and visualization let us plot confusion matrices.
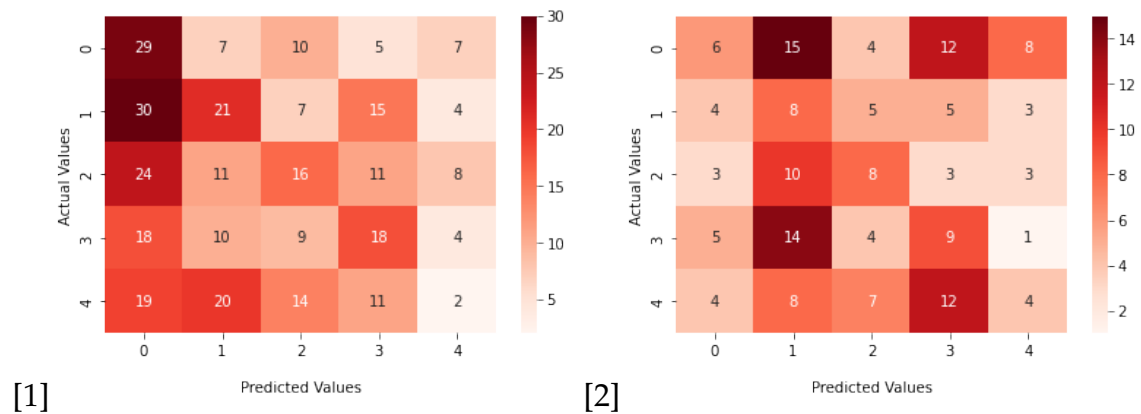

[1]   [2]

Figure 4.4: Confusion matrix of KNN on preprocessing attempts 1 & 2

As we can observe, the difference is not so significant.

Chapter 5

Conclusion and Further work

5.1   Conclusion

Diabetic Retinopathy is a growing hazard for public health. Unmonitored high levels of blood sugar might lead to DR, moreover, to blindness. Thus, we strongly believe that early detection of the disease will help prevent vision loss. Applying machine learning techniques and neural networks for detection of DR might make the diagnosis of the diseases more precise.

Unfortunately, the classical machine learning algorithms did not perform well on our preprocessing techniques. As opposed to this, the usage of Convolutional Neural Network, namely pretrained Resnet-34 proved to be pretty promising for our specific problem, giving an accuracy of 85%.

5.2   Further Work

As a further work, we are planning to increase the accuarcy of Resnet-34 by training on preprocessed images. In addition to this, as diabetic retinopathy is diagnosed by paying a specific attention to not only blood vessel, but also hard exudates and hemorrhages, we are planning to apply feature extraction techniques, to extract those features and give as an input images of extracted features on the same models.

5.3   Acknowledgements

Bibliography

[Cli00]   Mayo Clinic. *Diabetic retinopathy*. 2000.

[Gro04]   The Eye Diseases Prevalence Research Group. "The Prevalence of Diabetic Retinopathy Among Adults in the United States". In: *Archives of Ophthalmology* 122.4 (2004), pp. 552–563.

[PA13]   R. Priya and P. Aruna. "Diagnosis of Diabetic Retinopathy Using Machine Learning Techniques". In: (2013).

[Ram18]   V. Ramya. "SVM Based Detection for Diabetic Retinopathy". In: *International Journal of Research and Scientific Innovation* V (2018).

[P.S19]   et. al P.Sureka. "Diabetic Retinopathy Classification using FP Growth Algorithm and Probabilistic Neural Network". In: *SSRG International Journal of Computer Science and Engineering* (2019).

[Sci20]   Towards Data Science. *FP Growth Algorithm*. 2020.

[Neo21]   Aman Neo. *Diabetic Retinopathy Arranged*. 2021.

[Dyn19]   DynamSoft. *Image Processing 101 Chapter 1.3: Color Space Conversion*. 2019.

[Ros21]   Adrian Rosebrock. *OpenCV Histogram Equalization and Adaptive Histogram Equalization (CLAHE)*. 2021.

[Hos20]   Mehdi Hosseinzadeh. "Robust control applications in biomedical engineering: Control of depth of hypnosis". In: (2020).

[LSA17]   et. al. Lambros S. Athanasiou. "Plaque Characterization Methods Using Optical Coherence Tomography". In: (2017).

[Vis]   Open Source Computer Vision. *Adaptive Thresholding*.

[A.M20]   S. Seth A.Murzova. *What is Image Thresholding?:Otsu's Thresholding*. 2020.

[Ots79]   Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.

[ICA18]   et. al Ivan Cruz-Aceves. "A Novel Multiscale Gaussian-Matched Filter Using Neural Networks for the Segmentation of X-Ray Coronary Angiograms". In: *Journal of Healthcare Engineering* 2018 (2018).

[MAN17]   ARSHAD G ROHAN S SIDDHARTH KUMAR IYER PRAKKASH MANOHAR. *Diabetic Retinopathy Detection Using Python*. https : / / github . com / rsk97 / Diabetic - Retinopathy - Detection / blob / master / Diabetic _ retinopathy_detection.ipynb. 2017.