

BM 59D Final Project

Diabetic Retinopathy Detection

Kaan Oktay
2016701108

Alper Bayram
2016701024

1 Introduction

Diabetic Retinopathy (DR) is an eye disease associated with long-standing diabetes and the leading cause of blindness in the working-age population of the developed world. Around 40% to 45% of Americans with diabetes have some stage of the disease. Currently, detecting DR is a time-consuming and manual process that requires a trained clinician to examine and evaluate digital color fundus photographs of the retina. In recent years, there are multiple efforts to automatize this manual process [1–4].

In this project, we will try to classify the level of diabetic retinopathy of each patient into 5 different disease stages using a dataset published by Kaggle¹. For this purpose, Python programming language was used for coding the image processing algorithms and implementing classification methods. In Python, we used the following external open-source libraries for their built-in machine learning and image processing functions: NumPy, Pandas, OpenCV, Matplotlib, SciPy, SciKitImage, PyWavelets, SciKitLearn and Tabulate.

2 Data

We are provided with a large dataset of high-resolution retina images taken under various imaging conditions. A left and right field is provided for every patient. Each image is labelled with a subject ID as well as either left or right. A clinician has already rated the stage of diabetic retinopathy in each image on a scale of 0 to 4. Severity of the diabetic retinopathy increases as the value of scale increases. We formed our dataset with 700 images from each stage for this project.

Each image in the dataset had been recorded with different conditions. Also some of them had been recorded anatomically inverted or with different angles. In addition, images may contain artifacts. Another difficulty before processing the data is that there are lots of image and all of them have a really high resolution. To achieve a fast feature extraction and an accurate classification, we should apply some pre-processing methods to our data.

3 Pre-Processing

Pre-processing is vital to ensure that the dataset is consistent and displays only relevant features. An example of dataset images can be seen in Fig. 1.

As a common step for all feature extraction methods, dataset images were resized to 512x512 pixels by downsampling. This was done to make feature extraction faster. Other than resizing, some other pre-processing methods like applying contrast limited adaptive histogram equalization (CLAHE) and isolating green channel from the RGB image are used for some but not all extraction methods. These will be covered in the next section.

¹<https://www.kaggle.com/c/diabetic-retinopathy-detection/data>

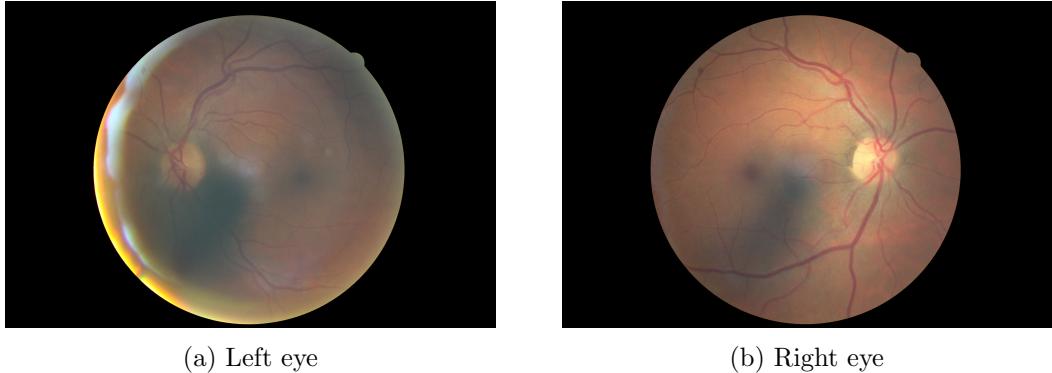


Figure 1: An example of dataset images.

4 Feature Extraction

In the literature, there many features proposed to detect diabetic retinopathy from digital fundus images. Visualization of significant eye features which are commonly used in literature for the diagnosis of diabetic retinopathy can be seen in Fig. 2.

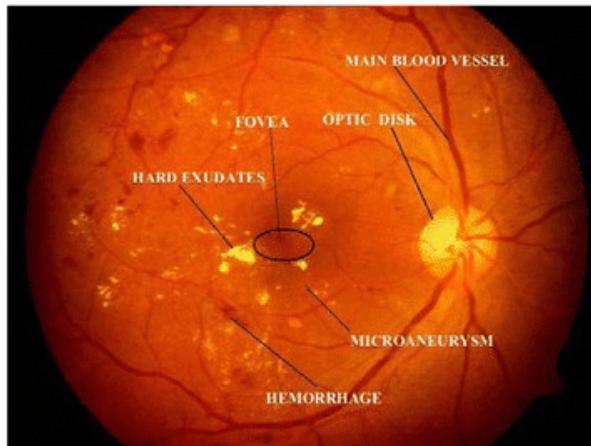


Figure 2: Important features of eye images for diagnosis.

The following features were chosen and extracted from images to detect diabetic retinopathy for our project: Shannon entropy, optic distance and blood vessel properties. In total, we have 6 different features extracted for each image.

4.1 Shannon Entropy

Shannon entropy is the expected average value of the information contained in each image. We expect a deferred entropy value at the images of higher stages of disease compared to the healthy images. For the calculation of entropy, just resized images were used without applying any further pre-processing technique.

4.2 Optic Distance

This feature is the distance between the center of the eye and the optic nerve head (the optic disc) where the optic nerve fibers leave the retina. To detect the optic nerve head, we used the fact that it

is the largest circular part among the brightest parts of the image and we should find the center of this part.

For this feature, in addition to resizing, CLAHE is applied to the image as a pre-processing method. The bright spot is usually yellow coloured so it should be bright in red and green channels. We isolated these channels from RGB images and then applied median filtering to eliminate unwanted bright parts. Then "bitwise and" operator was applied to the isolated images to remove the unwanted spots from both the channels and after that, median filtering was applied again. After getting rid of unwanted bright portions, we located the extreme points in the image to mask out the optical disc. Finally, the largest circular part was found and the distance from its center to the center of the eye was calculated. Detection of the largest, circular and brightest part of an eye image can be visualized as Fig. 3.

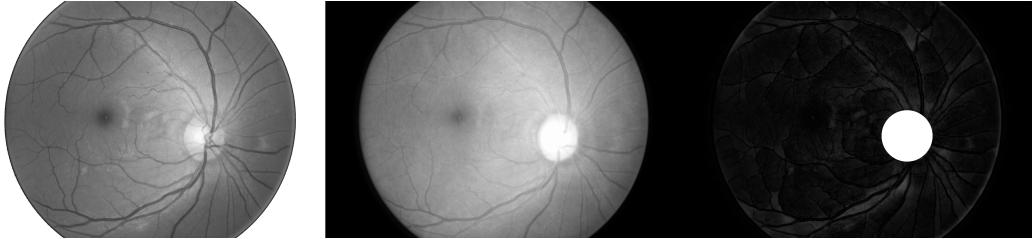


Figure 3: From left to right: green channel image, red channel image and spotted optic disc.

4.3 Blood Vessels

For diagnosis of diabetic retinopathy, blood vessels may carry significant information in their properties like area, thickness etc. We are trying to extract blood vessels and their properties from eye images to form a new feature for classification.

Firstly, we isolated green channel from RGB images as a pre-processing step because that provides the most contrast between blood vessels and the other parts of eye. After pre-processing, morphological opening and closing operations were done on the images to remove noise. Then CLAHE was applied to the images. After that, we removed very small contours through area parameter noise removal. Finally, we tried to extract blood vessel regions by removing blobs of unwanted bigger chunks taking in consideration that they are not straight lines like blood vessels.

Four properties of blood vessels were used as distinct features for classification of each image. These are mean of blood vessel thicknesses, max blood vessel thickness, number of blood vessels and total area of blood vessels for each eye image. Visualization of extracted blood vessels for an eye image can be seen in Fig. 4.



Figure 4: From left to right: green channel image, histogram equalized image and blood vessels.

5 Outlier Detection

For the outlier detection, we used Mahalanobis distance metric. For each image, if one of the features has a Mahalanobis distance to the mean of the feature exceeds the threshold, that image was excluded for next steps. Threshold value was determined as 10 for this project.

6 Feature Selection

For the feature selection, we calculated f-scores for each feature. Then we applied our classifiers for the subsets of features with 4, 5 and 6 maximum f-scores to compare. When we calculated the average accuracy for each subset, the most successful one was the subset that we used all the features. Therefore, we did not exclude any features from next steps.

7 Classification and Results

Remaining features and images from previous steps were used in this step for classification. In order to evaluate classification performance, we used quadratic weighted kappa which measures the agreement between two ratings. This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters). In the event that there is less agreement between the raters than expected by chance, this metric may go below 0. The quadratic weighted kappa is calculated between the scores assigned by the human rater and the predicted scores.

Images have five possible ratings 0, 1, 2, 3, and 4. Each image is characterized by a tuple (ea,eb), which corresponds to its scores by rater A (human) and rater B (predicted). The quadratic weighted kappa is calculated as follows. First, an N-by-N histogram matrix O is constructed, such that O_{ij} corresponds to the number of images that received a rating i by A and a rating j by B. An N-by-N matrix of weights, w, is calculated based on the difference between raters' scores:

$$w_{i,j} = \frac{(i - j)^2}{(N - 1)^2}$$

An N-by-N histogram matrix of expected ratings, E , is calculated, assuming that there is no correlation between rating scores. This is calculated as the outer product between each rater's histogram vector of ratings, normalized such that E and O have the same sum.

From these three matrices, the quadratic weighted kappa was calculated as follows. A built-in function for calculating κ was used from the libraries previously mentioned.

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}$$

Parameters of SVM classifiers were optimized by their own SciKit-Learn functions. Also for kNN, k is chosen as 30. Confusion matrices and respective κ scores for all classifiers can be seen in the Fig. 5, 6, 7, 8, 9, 10, and 11. All κ scores were calculated by using 10-fold cross-validation.

8 Discussion

As can be seen, kNN classifier is the most successful one with $\kappa = 0.35$. We are especially good at accurate classification of the most severe disease stage. When we compare our best score with the Kaggle Leaderboard² scores for this competition, our rank is 150th among 661 competitors. It is fairly good

²<https://www.kaggle.com/c/diabetic-retinopathy-detection/leaderboard>

because we have a small set of extracted features, only traditional classifiers and lots of artifacts in images. All the highest rank competitors used state-of-the-art machine learning methods like convolutional neural networks or recurrent neural networks for classification and also feature extraction. Our Python code for this project can be found in our GitHub repository³.

References

- [1] Bhuiyan, A., Nath, B., Chua, J., and Kotagiri, R., Blood vessel segmentation from color retinal images using unsupervised texture classification. IEEE Int. Conf. Image Processing, ICIP 5:521524, 2007
- [2] Acharya, U. R., Lim, C. M., Ng, E. Y. K., Chee, C., and Tamura, T., Computer based detection of diabetes retinopathy stages using digital fundus images. J. Eng. Med. 223(H5):545553, 2009.
- [3] Hunter, A., Lowell, J., Owens, J., and Kennedy, L, Quantification of diabetic retinopathy using neural networks and sensitivity analysis, In Proceedings of Artificial Neural Networks in Medicine and Biology, pp. 81-86, 2000.
- [4] Sinthanayothin, C., Boyce, J. F., Williamson, T. H., and Cook, H. L., Automated detection of diabetic retinopathy on digital fundus image. Diabet. Med. 19(2):105112, 2002.
- [5] Sisodia, D. S., Nair, S., and Khobragade, P., Diabetic Retinal Fundus Images: Preprocessing and Feature Extraction for Early Detection of Diabetic Retinopathy. Biomedical and Pharmacology Journal, 10(2): 615-626 (2017).

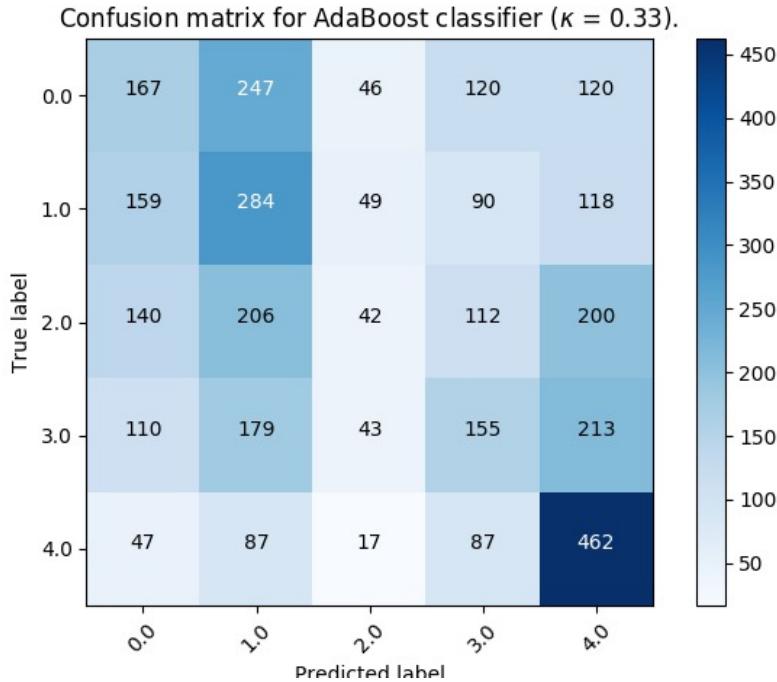


Figure 5: Confusion matrix and κ score for AdaBoost classifier.

³<https://github.com/alpby/ProjectRetina.git>

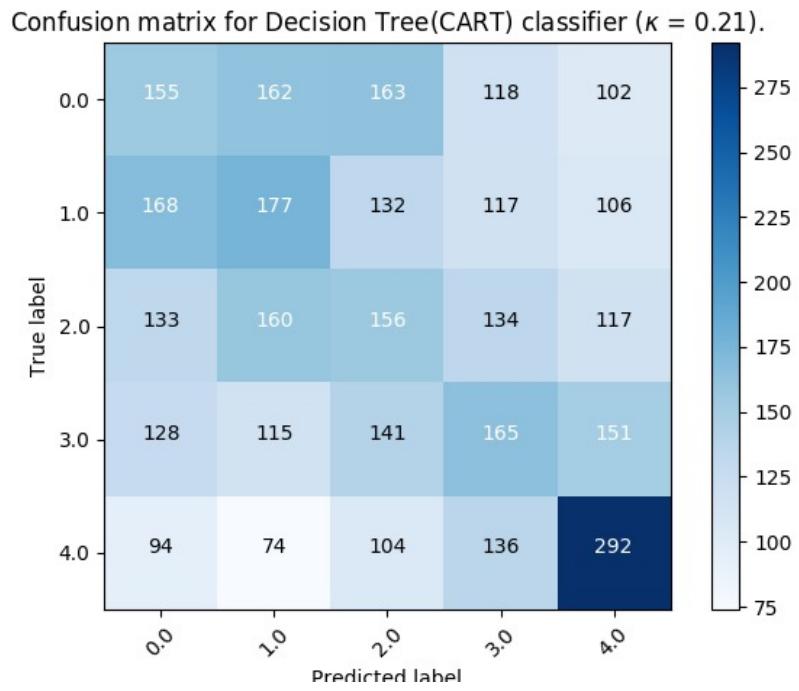


Figure 6: Confusion matrix and κ score for Decision Tree classifier.

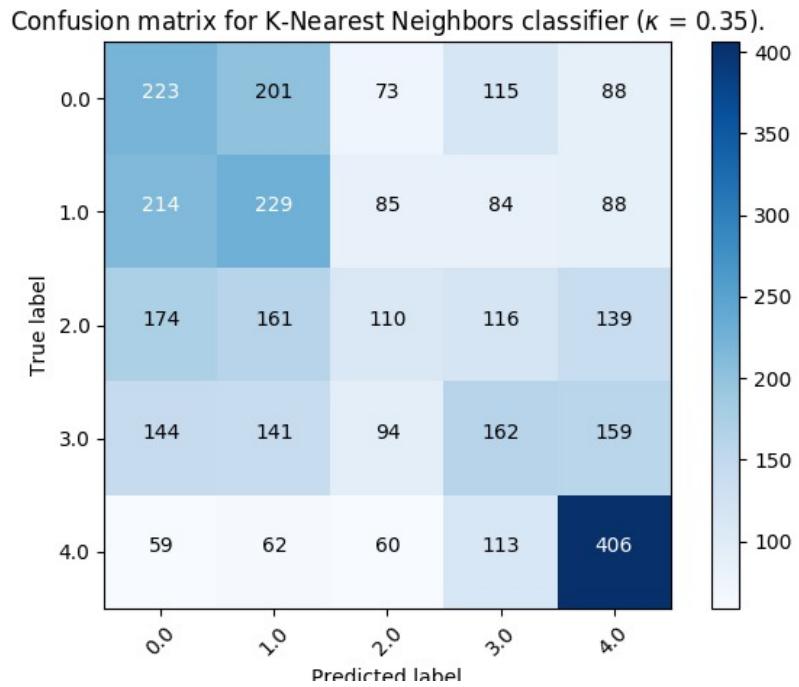


Figure 7: Confusion matrix and κ score for kNN ($k=30$) classifier.

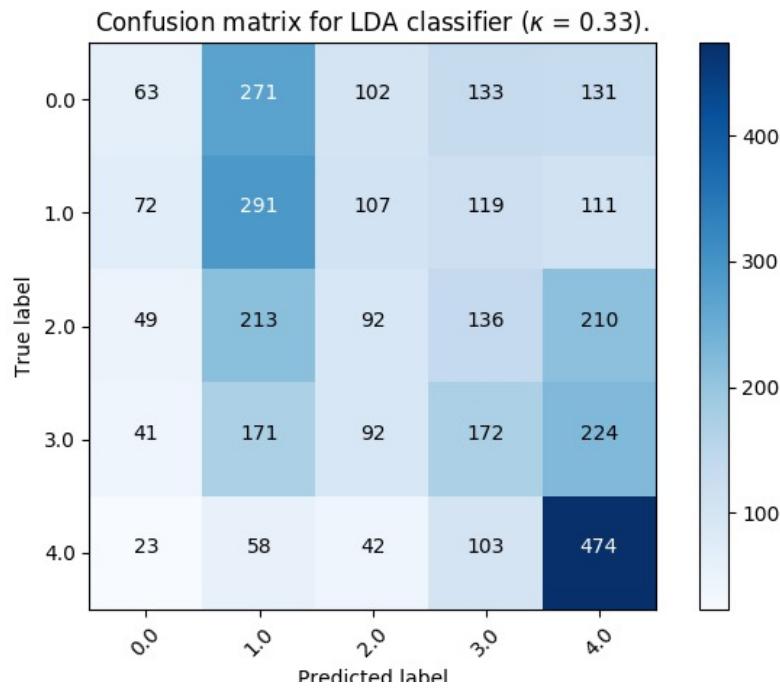


Figure 8: Confusion matrix and κ score for Linear Discriminant Analysis classifier.

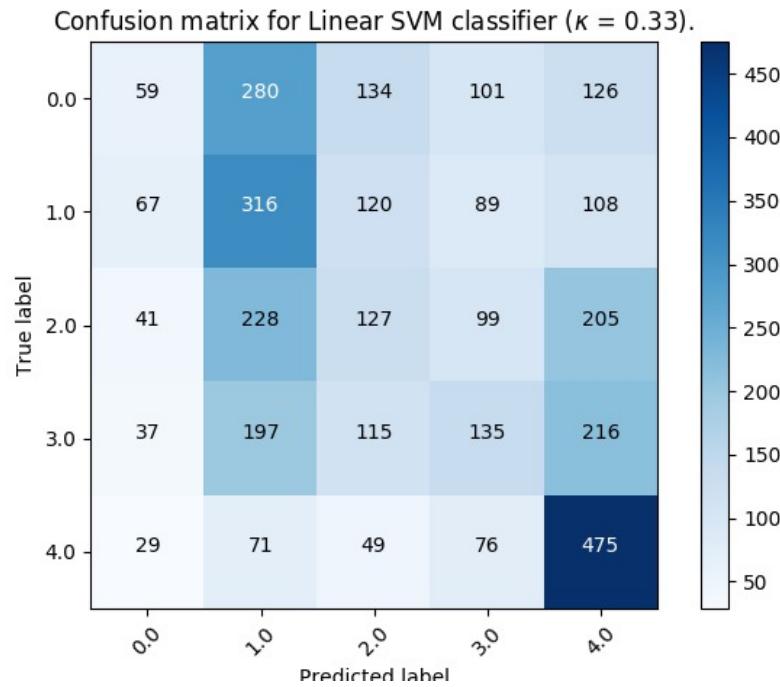


Figure 9: Confusion matrix and κ score for SVM (linear kernel) classifier.

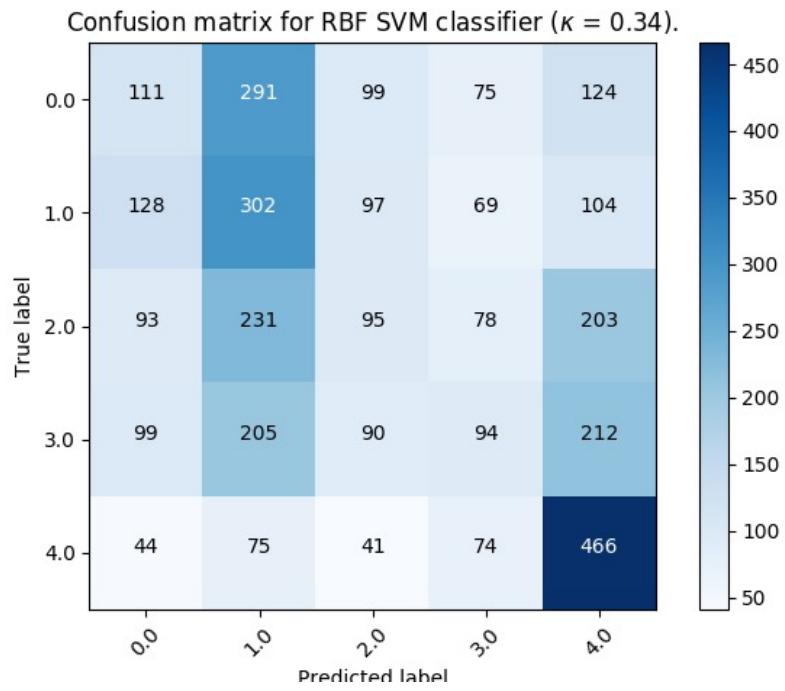


Figure 10: Confusion matrix and κ score for SVM (RBF kernel) classifier.

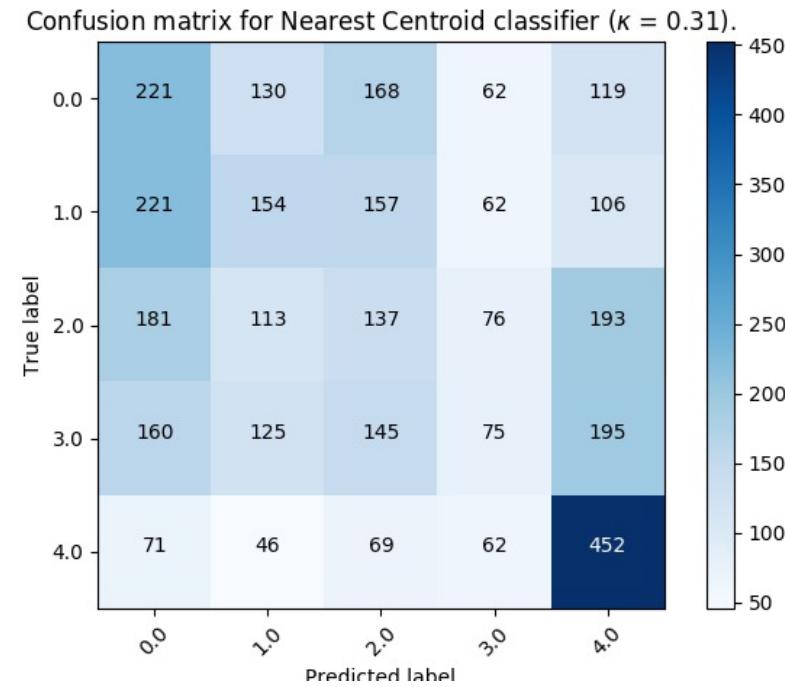


Figure 11: Confusion matrix and κ score for Nearest Centroid classifier.