MATH 637 – MATH TECHNIQUES IN DATA SCIENCE
FINAL PROJECT
DIABETES PREDICTION AND DIABETIC RETINOPATHY DETECTION

Team:

1. Sri Vishnuvardhan Reddy Akepati - akepati@udel.edu
2. Venkata Sai Vardhan Kataru - vardhan@udel.edu
3. Nagendra Sai Nandimandalam - nagendra@udel.edu
4. Charitha Nagamalla - shettych@udel.edu
5. Sujith Yeluru - sujith@udel.edu

ABSTRACT:

Diabetes is a global epidemic, affecting millions and leading to serious complications. In our machine learning project, we aim to tackle two critical objectives: predicting diabetes (binary) and detecting diabetic retinopathy (multi-class). Through cutting-edge algorithms and deep learning techniques, we achieve high accuracy in both classifications, analyzing medical records and retinal images to inform clinical decision-making and improve patient outcomes. Our results pave the way for more accurate and effective tools for early detection and treatment, addressing two of the most pressing challenges in diabetes management and screening.

Keywords: Deep Learning, Diabetes, Binary classification, multi-class classification.

Table of Contents

Part-1: Diabetes Prediction

1.0 INTRODUCTION: -

Diabetes, a chronic metabolic disorder affecting millions worldwide, remains a pressing global health issue. With its debilitating complications and increasing prevalence, accurate prediction of diabetes has become crucial for early intervention and effective management. In this report, we explore the captivating field of binary classification for diabetes prediction, focusing on innovative techniques and advancements that enable the identification of individuals at risk with remarkable precision.

Advancements in machine learning have revolutionized healthcare, empowering researchers and healthcare professionals to develop robust predictive models for diabetes detection. By leveraging a wide range of features such as clinical measurements, genetic markers, and lifestyle factors, these models can classify individuals into two categories: those at high risk of developing diabetes and those at low risk. This binary classification approach enables healthcare providers to implement proactive measures, including personalized interventions, lifestyle modifications, and targeted preventive strategies. Through this report, we aim to provide a comprehensive overview of the methodologies employed in diabetes prediction, highlighting their potential to transform healthcare by enabling early detection and improving the overall management of this prevalent condition.

1.1 DATASET DESCRIPTION: -

The Behavioral Risk Factor Surveillance System (BRFSS), a CDC annual survey, collects data on health-related behaviors, chronic conditions, and preventative service use from over 400,000 Americans. For this project, a dataset processed from the original dataset was used.

The dataset analyzed contains 253,680 instances with 22 features. These features encompass a wide range of patient information, such as their BMI, whether they have high blood pressure or high cholesterol, their fruit and vegetable consumption, and more. The target variable is a binary representation of whether the individual has diabetes or not.

1.1.1 VARIABLE DESCRIPTION: -

1. **Diabetes_012: Represents diabetes status.**
   - 0 = no diabetes
   - 1 = diabetes
2. **HighBP: Indicates high blood pressure status.**
   - 0 = no high BP
   - 1 = high BP

3. **HighChol: Indicates high cholesterol status.**
   - 0 = no high cholesterol
   - 1 = high cholesterol

4. **CholCheck: Indicates if cholesterol was checked in the past 5 years.**
   - 0 = no cholesterol check in 5 years
   - 1 = yes, cholesterol check in 5 years

5. **BMI: Body Mass Index.**
   BMI is a crucial feature. It is well-established in medical research that a higher BMI can be associated with an increased risk of several health conditions, including diabetes.

6. **Smoker: Indicates if individual has smoked at least 100 cigarettes in their entire life.**
   - 0 = no
   - 1 = yes

7. **Stroke: Indicates if individual has ever been told they had a stroke.**
   - 0 = no
   - 1 = yes

8. **HeartDiseaseorAttack: Indicates the presence of coronary heart disease (CHD) or myocardial infarction (MI).**
   - 0 = no
   - 1 = yes

9. **PhysActivity: Indicates physical activity in the past 30 days not including job.**
   - 0 = no
   - 1 = yes

10. **Fruits: Indicates daily fruit consumption**.
    - 0 = no
    - 1 = yes

11. **Veggies: Indicates daily vegetable consumption.**
    - 0 = no
    - 1 = yes

12. **HvyAlcoholConsump: Indicates if individual is a heavy drinker.**
    - 0 = no
    - 1 = yes

13. **AnyHealthcare: Indicates if individual has any kind of health care coverage.**
    - 0 = no
    - 1 = yes

14. **NoDocbcCost: Indicates if there was a time in the past 12 months when an individual needed to see a doctor but could not be due to cost.**
    - 0 = no
    - 1 = yes

15. **GenHlth: General health status scale (1-5).**
    - 1 = excellent
    - 2 = very good
    - 3 = good
    - 4 = fair
    - 5 = poor

16. **MentHlth: Number of days in past 30 days mental health was not good (1-30 days).**

17. **PhysHlth: Number of days in past 30 days physical health was not good (1-30 days).**

18. **DiffWalk: Indicates if individual has serious difficulty walking or climbing stairs.**
    - 0 = no
    - 1 = yes

19. **Sex: Gender of the individual.**
    - 0 = female
    - 1 = male

20. **Age: 13-level age category (_AGEG5YR).**

21. **Education: Education level (EDUCA).**
    - 1 = Never attended school or only kindergarten
    - 2 = Grades 1 through 8 (Elementary)
    - 3 = Grades 9 through 11 (Some high school)
    - 4 = Grade 12 or GED (High school graduate)
    - 5 = College 1 year to 3 years (Some college or technical school)
    - 6 = College 4 years or more (College graduate)

22. **Income: Income scale (INCOME2).**
    - 1 = Less than $10,000
    - 2 = Between $10,000 and less than $15,000
    - 3 = Between $15,000 and less than $20,000
    - 4 = Between $20,000 and less than $25,000
    - 5 = Between $25,000 and less than $35,000
    - 6 = Between $35,000 and less than $50,000
    - 7 = Between $50,000 and less than $75,000
    - 8 = $75,000 or more

1.2 PROBLEMS WORKED: -

The focus of this project lies in identifying the presence of diabetes using various machine learning algorithms. We encounter an issue of imbalance in the dataset, with far fewer instances of diagnosed diabetes than non-diabetes instances, which poses challenges in training effective machine learning models. We developed a diabetes prediction web app using the Streamlit Python web framework. This app aims to provide users with a convenient way to utilize our model for diabetes prediction.

1.3 PREPROCESSING: -

a. Checking for missing values: We don't have any missing values in our dataset.

b. Removing Duplicates: We found 24206 duplicate rows out of 253680 rows. We dropped them to make our dataset with unique columns.

c. Balancing Dataset: We used isolation forest from sklearn.ensemble to identify the outliers/anomalies. Since we have a large amount of dataset, we dropped the outliers instead of adjusting to the dataset.

d. Exploring Categorical columns: We plot the binary columns with respect to diabetes to observe the distribution of data. The plot is shown in fig 1.1.

e. Exploring numerical columns: We plot the histograms of each numerical column to observe the distribution of data. The plot is shown in fig 1.2.

f. Heatmap: We plotted a histogram to observe the correlations between each variable. The plot is shown in fig 1.3. The observations from the heatmap shown a positive correlation between the Physical Health and General Health variables, suggesting that General Health may be a composite measure comprising Physical Health Sum and other related factors. Also, the analysis indicated a negative correlation between Income and General Health, implying that individuals with lower incomes may experience limited access to private medical care, potentially leading to negative impacts on their overall health outcomes.

g. Feature Selection: We plotted a bar plot that shows relations with diabetes column which is shown in fig 1.4. We dropped the columns that indicate neither positive nor negative relation with diabetes column. These variables have more correlated with the target variable Diabetes_binary, GenHlth, HighBP, DiffWalk, BMI, HighChol, Age, HeartDiseaseorAttack, PhysHlth, Physactivity, Education, Income. These variables that have a very weak correlation AnyHealthcare, NoDocbcCost, Fruits, Sex, Smoker, Veggies.

h. Scaling and Splitting: The resultant data from the above techniques has been scaled using Standard Scaler. Since we are dealing with different features that have different units and scales such as 'Age', 'BMI', 'Income' etc. To ensure that these different scales do not influence the model's performance, you've standardized these features to have a mean of 0 and a standard deviation of 1. Later, we split the dataset into a training set and a test set using the train_test_split taking 20% as test data. The training data is used to train the model while the test data is used to evaluate the model's performance.

1.4 METHODS and PARAMETERS: -

The analysis uses multiple supervised learning techniques are Logistic Regression, Gaussian Naive Bayes, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting and XGBoost. The methods have been chosen due to their versatility and efficacy in binary classification tasks. A brief literature review follows.

- **Logistic Regression:**

  In the realm of binary classification for diabetes prediction, one widely employed algorithm is Logistic Regression, available in the popular scikit-learn library. Logistic Regression is a linear classifier that models the relationship between input features and the probability of belonging to a particular class. It is particularly well-suited for binary classification tasks, such as distinguishing individuals at high risk of developing diabetes from those at low risk.

  When utilizing Logistic Regression in scikit-learn, various parameters are tuned to optimize the model's performance. They are the lr_param_grid parameter grid offers a range of options to fine-tune the model. The 'C' parameter, which controls the inverse of regularization strength, can take values of 0.01, 0.1, and 1. Regularization helps prevent overfitting and balances the trade-off between fitting the training data and generalizing to unseen data. Additionally, the 'penalty' parameter specifies the type of regularization, with options for 'l1' (L1 regularization, also known as Lasso) and 'l2' (L2 regularization, also known as Ridge). The 'solver' parameter determines the algorithm used in the optimization problem, with choices including 'newton-cg', 'lbfgs', 'liblinear', 'sag', and 'saga'. Each solver has its own advantages and is suited for different types of datasets. By carefully selecting and optimizing these parameters, Logistic Regression can be effectively utilized to create accurate and reliable models for diabetes prediction.

- **Gaussian Naive Bayes:**

  Another popular algorithm for binary classification in diabetes prediction is Gaussian Naive Bayes, available in the scikit-learn library. Gaussian Naive Bayes is a probabilistic classifier that assumes the features follow a Gaussian distribution. It calculates the posterior probability of each class given the input features and selects the class with the highest probability. This algorithm is known for its simplicity, efficiency, and effectiveness in handling high-dimensional data.

  When using Gaussian Naive Bayes in scikit-learn, one key parameter that adjusted is 'var_smoothing'. This parameter represents the portion of the largest variance of all features that is added to variances for calculation stability. The nb_param_grid parameter grid specifies a single option for 'var_smoothing' with a value of 1e-09. By adjusting this parameter, we fine-tune the model's performance and improve its accuracy in predicting diabetes outcomes. Gaussian Naive Bayes, with its simplicity and efficient implementation, offers a reliable approach for binary classification in diabetes prediction tasks, providing valuable insights for healthcare professionals in identifying individuals at risk and guiding appropriate interventions.

- **K-Nearest Neighbors:**

  K-Nearest Neighbors (KNN) is a versatile algorithm widely employed for binary classification in diabetes prediction, leveraging the scikit-learn library. KNN is a non-parametric method that classifies data points based on their proximity to neighboring points in the feature space. When utilizing KNN in scikit-learn, key parameters that adjusted to enhance the model's performance. The knn_param_grid parameter grid includes options for 'n_neighbors' (the number of nearest neighbors to consider), with values of 3, 5, and 7, influencing the model's sensitivity to local patterns. The 'weights' parameter allowed us to choose between 'uniform' (giving equal weight to all neighbors) and 'distance' (giving higher weight to closer neighbors). Additionally, the 'algorithm' parameter specifies the algorithm used to compute nearest neighbors, with choices including 'ball_tree' and 'kd_tree'. By carefully tuning these parameters, KNN can effectively identify patterns in diabetes data and classify individuals into high-risk or low-risk categories, aiding in early intervention and personalized healthcare strategies.

- **Decision Tree:**

  In the context of binary classification for diabetes prediction, Decision Trees have proven to be a powerful and interpretable algorithm, widely employed for their ability to capture complex relationships between features and target classes. The scikit-learn library provides a flexible implementation of Decision Trees, allowing us to fine-tune their performance through parameter adjustments. The dc_param_grid parameter grid includes various parameters to optimize the model, such as 'max_depth' which determines the maximum depth of the tree, 'criterion' specifying the quality measure used for splitting ('gini' for Gini impurity or 'entropy' for information gain), 'min_samples_split' defining the minimum number of samples required to split an internal node, and 'min_samples_leaf' indicating the minimum number of samples required to be a leaf node. By exploring different combinations of these parameters, researchers can find the optimal settings that produce accurate and reliable Decision Tree models for diabetes prediction, facilitating informed decision-making in healthcare interventions.

- **Random Forest:**

  Random Forest, a powerful ensemble learning algorithm, proves to be highly effective for binary classification in diabetes prediction when implemented using the scikit-learn library. By constructing an ensemble of decision trees, Random Forest combines their predictions to make robust and accurate classifications. In this context, the rf_param_grid parameter grid specifies key parameters for optimization. The 'n_estimators' parameter determines the number of decision trees to include in the forest, with options of 50, 100, and 200. The 'max_depth' parameter limits the maximum depth of each decision tree, controlling the model's complexity and potential overfitting. It is set to values of 3, 5, and 7 in this case. Additionally, the 'min_samples_split' parameter specifies the minimum number of samples required to split an internal node and affects the tree's branching process.

- **Gradient Boosting:**

  Gradient Boosting is a powerful machine learning algorithm widely used for binary classification, including in the context of diabetes prediction. The algorithm, available in the scikit-learn library, constructs an ensemble of weak prediction models, typically decision trees, and combines them iteratively to create a strong predictive model. This iterative process focuses on learning from the mistakes of previous models, gradually improving the overall prediction accuracy. In the given example, the gb_param_grid parameter grid provides options to fine-tune the Gradient Boosting model. The 'n_estimators' parameter determines the number of weak learners (decision trees) to be sequentially added to the ensemble, with choices of 50, 100, and 200. The 'learning_rate' parameter controls the contribution of each weak learner to the overall model, with values of 0.01, 0.1, and 1. A lower learning rate requires more iterations but may improve the model's generalization. The 'max_depth' parameter limits the maximum depth of each decision tree in the ensemble, with options of 3, 5, and 7. By adjusting these parameters, practitioners can optimize the Gradient Boosting model's performance for accurate binary classification in diabetes prediction, enhancing its ability to identify individuals at risk and facilitate timely interventions.

- **XGBoost:**

  XGBoost, an optimized gradient boosting algorithm, has gained considerable popularity in binary classification tasks, including diabetes prediction, and is readily available in the scikit-learn library. XGBoost excels at ensemble learning by combining the predictions of multiple weak learners, such as decision trees, into a strong predictive model. With its ability to handle complex relationships and high-dimensional data, XGBoost has become a powerful tool in machine learning. When using XGBoost in scikit-learn, we fine-tuned the model's performance by adjusting several parameters. They are the xgb_param_grid parameter grid specifies options for 'n_estimators' (the number of boosting iterations), 'learning_rate' (the step size shrinkage), 'max_depth' (the maximum depth of each tree), and 'eval_metric' (the evaluation metric to optimize, in this case, 'error'). By exploring different combinations of these parameters, healthcare professionals can harness the full potential of XGBoost to improve the accuracy of binary classification for diabetes prediction and aid in making informed decisions regarding patient management and intervention strategies.

1.5 RESULTS: -

Based on the results obtained from our experimentation with various classification models for diabetes prediction, it is evident that XGBoost outperformed other algorithms across multiple performance metrics. With a high training accuracy of 92.45%, an impressive training AUC of 0.7986, and a low training RMSE of 0.2747, XGBoost demonstrated its ability to effectively capture complex relationships within the data.

Also, XGBoost exhibited excellent performance on the test dataset, with an accuracy of 92.56% and an AUC of 0.7878. These results indicate the model's robustness and its capability to generalize well to unseen data. Additionally, the test RMSE of 0.2728 further demonstrates the model's accuracy in predicting diabetes outcomes.

The results of all the algorithms, their train, test accuracies and AUC score as shown in Table-1

Models	Train Accuracy	Test Accuracy	Test AUC
Logistic regression	92.3%	92.48%	0.769
Gaussian Naïve Bayes	74.9%	74.6%	0.664
K Nearest Neighbors	92.08%	91.8%	0.773
Decision Trees	92.45%	92.45%	0.730
Random Forest	92.45%	92.56%	0.758
Gradient Boosting	92.45%	92.56%	0.754
XGBoost	92.45%	92.57%	0.787

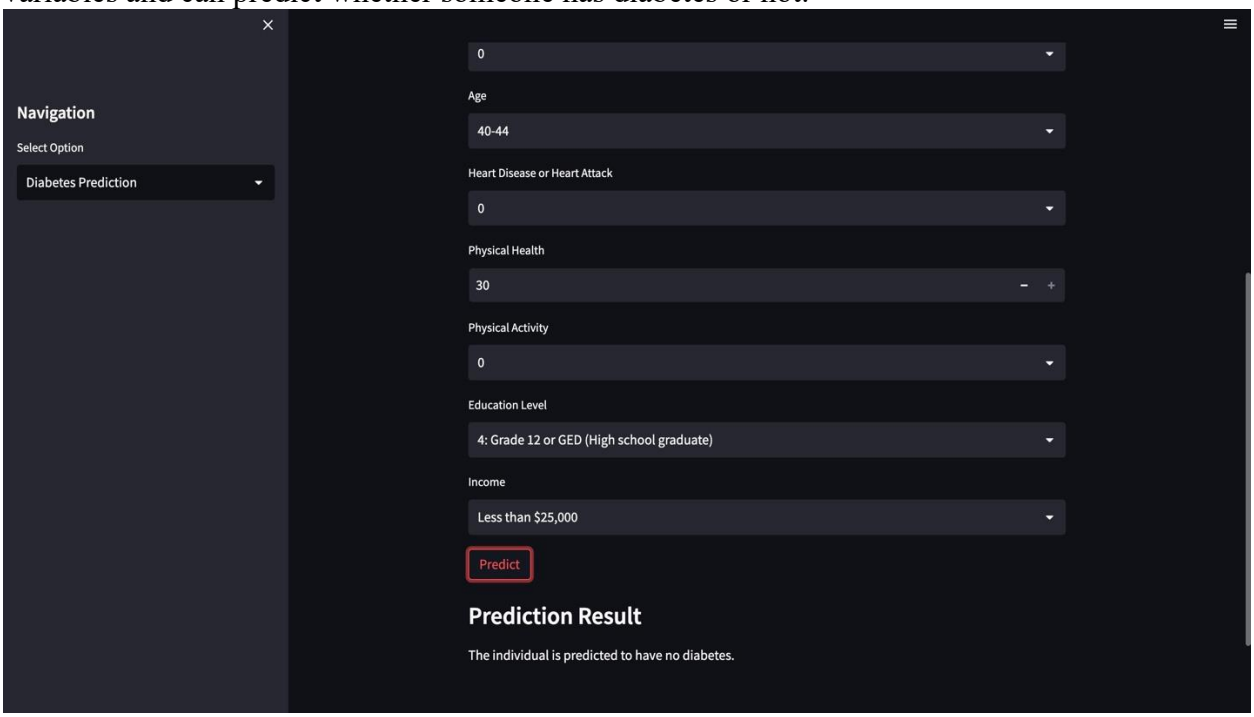
Table-1

1.6 DEPLOYMENT: -

We used pickle library to save the best performing algorithm (XGBoost Algorithm) and used it in the app. The trained models and scaler are loaded back into the application. The app defines a prediction function that scales the input data and uses the XGBoost Classifier to predict the likelihood of diabetes. It also includes a function to remove outliers from the input data using the Isolation Forest model. The app interface is designed with Streamlit, allowing users to input health parameters. When the 'Predict' button is clicked, the app calls the prediction function with user input, and displays whether the individual is predicted to have diabetes or not based on the given health information.

These are the screenshots of the app, The first one is the home page.

The following one is the prediction page where you can give values to various demographic variables and can predict whether someone has diabetes or not.



The trained models and scaler are loaded back into the application. The app defines a prediction function that scales the input data and uses the Gradient Boosting Classifier to predict the likelihood of diabetes. It also includes a function to remove outliers from the input data using the Isolation Forest model.

The app interface is designed with Streamlit, allowing users to input health parameters. When the 'Predict' button is clicked, the app calls the prediction function with user input, and displays whether the individual is predicted to have diabetes or not based on the given health information.

1.7 CONCLUSION: -

In conclusion, this project demonstrated the significance of machine learning models in predicting diabetes using health-related data. Among the various models employed, the Gradient Boosting Classifier displayed the best performance, achieving the highest accuracy. This study emphasizes the potential of utilizing sophisticated machine learning techniques for predictive healthcare analytics. Nevertheless, to further refine our models and enhance prediction outcomes, other techniques such as Synthetic Minority Over-sampling Technique (SMOTE) can be employed to address class imbalance in the dataset. Thus, continuous exploration and application of more advanced techniques are necessary for future work in this domain.

Part-2: Diabetic Retinopathy Detection

2.0 INTRODUCTION: -

Diabetic retinopathy, a progressive eye disease resulting from complications of diabetes, has emerged as a leading cause of vision loss and blindness worldwide. It affects individuals with diabetes, leading to damage in the blood vessels of the retina, the light-sensitive tissue at the back of the eye. Early detection and timely intervention are crucial in preventing irreversible vision impairment and improving patient outcomes. In this report, we delve into the critical field of diabetic retinopathy detection, exploring innovative techniques and advancements that have revolutionized the diagnosis and treatment of this debilitating condition.

With the advent of advanced imaging technologies and machine learning algorithms, the landscape of diabetic retinopathy detection has been transformed. These technologies enable the automated analysis of retinal images to identify the presence and severity of diabetic retinopathy. By leveraging deep learning models, computer vision techniques, and large annotated datasets, we tried to accurately classify retinal images, predict disease progression, and recommend appropriate interventions. This 2nd part of report aims to provide a comprehensive overview of the cutting-edge methodologies and tools used in diabetic retinopathy detection, highlighting their potential to revolutionize healthcare by enabling early diagnosis, personalized treatment plans, and ultimately preserving the vision of individuals affected by this sight-threatening complication of diabetes.

2.1 Dataset Description: -

We used 3622 images from a Kaggle Dataset specifically taken from 'APTOS2019 Blindness Detection' competition. The images are real-world dataset, which has noise in both the images and labels. Images may contain artifacts, be out of focus, underexposed, or overexposed. The images were gathered from multiple clinics using a variety of cameras over an extended period.

A clinician has rated each image for the severity of diabetic retinopathy on a scale of 0 to 4:

0 - No DR

1 - Mild

2 - Moderate

3 - Severe

4 - Proliferative DR

2.2 Preprocessing: -

The very first step in the preprocessing pipeline is resizing the images. All images are resized to a standard size of 256x256 pixels. This is essential because machine learning models require inputs of the same dimensions and reduce computational complexity. By resizing, we ensure that all images processed by the models have consistent dimensions, which is a prerequisite for most models. The resultant images are shown in fig 2.1.

Following the resizing, images are converted to grayscale. While color images are represented in three dimensions (Red, Green, Blue), grayscale images are two-dimensional. This reduction not only simplifies the computation but also, in some cases, helps to identify the important features easily compared to color images. The resultant images are shown in fig 2.2.

After converting to grayscale, a Gaussian blur is applied to smooth the images. This process significantly reduces high-frequency noise that may arise from the image acquisition process or other environmental factors during the photography of the retina. The Gaussian blur effectively minimizes these inconsistencies, allowing the model to focus on the important features. The image after applying gaussian blur is shown in fig 2.3.

The next step in the pipeline is circular cropping. Retinal images typically contain vital structures at the center, which are crucial for diagnosis. Circular cropping is applied to crop the image in a circular shape around the center, eliminating the outer regions while preserving the important parts of the image. This step aids in directing the machine learning model's focus towards the critical parts of the image, eliminating any unnecessary distractions. The image after applying both gaussian blur and circular crop is shown in fig 2.4.

Once the images are suitably cropped, they are rescaled to normalize the image pixel values. Specifically, each pixel value is divided by 255.0, converting the range of each channel in the RGB image from 0-255 to 0-1. This standardization step is crucial for deep learning models, as having a consistent scale for input data helps stabilize and expedite the model's learning during training.

The final preprocessing step involves applying an image augmentation technique. The eye image after augmentation is shown in fig 2.5. Image augmentation employs various transformations, such as rotations, translations, and zooming, to generate additional synthetic training images. This technique effectively increases the size of the training dataset and introduces a form of regularization, making the model more robust against overfitting by addressing the problem of imbalance in dataset.

2.3 Algorithms and Implementation: -

In the quest to detect and classify diabetic retinopathy, a range of traditional machine learning algorithms, including Naive Bayes, KNN, SVM, Decision Tree, Random Forest, Gradient Boosting, XGBoost, CatBoost, and AdaBoost, have been employed. However, despite their widespread use in various domains, these algorithms have exhibited suboptimal performance in accurately identifying and classifying the severity of diabetic retinopathy. Consequently, an alternative approach utilizing deep learning algorithms has gained prominence, enabling significant improvements in detection accuracy, and paving the way for enhanced diagnosis and treatment of this vision-threatening condition.

The limitations of traditional machine learning algorithms in the context of diabetic retinopathy detection have prompted us to explore the potential of deep learning. Deep learning algorithms, such as convolutional neural networks (CNNs) and pre-trained networks have shown remarkable capabilities in handling complex patterns and features within retinal images. By leveraging deep neural networks with multiple layers of interconnected neurons, these algorithms can automatically extract intricate features from the images, enabling more accurate classification and prediction of diabetic retinopathy severity levels. The shift towards deep learning in diabetic retinopathy detection signifies a paradigm shift towards more sophisticated and data-driven approaches, focusing on optimizing neural network architectures, improving model performance, and ultimately revolutionizing the field of ophthalmology by enhancing the accuracy and efficiency of diabetic retinopathy diagnosis and treatment.

- ✓ **Artificial Neural Network:**

  The artificial neural network model we utilized for diabetic retinopathy detection demonstrates a sequential architecture. Comprising three dense layers, the model begins with a dense layer of 128 neurons using the ReLU activation function, followed by another dense layer of 64 neurons also utilizing the ReLU activation. The final layer consists of 5 neurons with the softmax activation function, enabling the classification of diabetic retinopathy severity into five levels. The model is trained using the Adam optimizer, employing the sparse categorical cross-entropy loss function to optimize performance. Training is conducted for 100 epochs with a batch size of 32. Following training, the model generates predictions for the test dataset, and the accuracy of the model is evaluated using the accuracy_score metric.

- ✓ **Convolutional Neural Network:**

  The convolutional neural network (CNN) model employed for diabetic retinopathy detection is designed with dropouts and L2 regularizes to enhance its performance and mitigate overfitting. The model architecture consists of multiple layers, beginning with a Conv2D layer with 32 filters and a kernel size of (3, 3). This layer applies the rectified linear unit (ReLU) activation function and includes an L2 regularization term of 0.01 to promote better generalization. MaxPooling2D layers with a pool size of (2, 2) follow each Conv2D layer to down sample the feature maps. Dropout layers with a rate of 0.25 are introduced to randomly deactivate 25% of the neurons, reducing the risk of overfitting. Additional Conv2D layers with 64 and 128 filters are included to extract more complex features. The final Conv2D

layer is followed by a Flatten layer to convert the output into a one-dimensional vector. Dense layers with 128 neurons and ReLU activation, along with dropout and L2 regularization, further capture high-level representations. Lastly, a Dense layer with softmax activation is used to produce the final classification output, indicating the severity level of diabetic retinopathy among five classes.

- ✓ **Transfer Learning with Inception V3:**

  The InceptionV3 model, utilized in this study for diabetic retinopathy detection, is a powerful deep learning architecture that has shown good performance in image classification tasks. Pre-trained on a large-scale dataset, InceptionV3 employs a combination of convolutional layers with various kernel sizes to capture multi-scale features effectively. By utilizing parallel branches with different filter sizes, InceptionV3 efficiently processes complex visual patterns, providing a rich representation of the input images. InceptionV3 also incorporates advanced techniques like batch normalization and factorized convolutions, which contribute to its exceptional accuracy and computational efficiency.

  The model built for diabetic retinopathy detection utilizes the InceptionV3 architecture as its base, with specific modifications tailored to the task at hand. Dropout layers were strategically inserted to mitigate overfitting, allowing the model to generalize well to unseen data. Dense layers with ReLU activation were added to capture higher-level representations, enhancing the model's ability to learn complex patterns within retinal images. By training the model in a two-phase approach, starting with frozen lower layers, and gradually fine-tuning the upper layers, it was possible to leverage the pre-trained weights while adapting the model to the specific requirements of diabetic retinopathy detection. Early stopping and learning rate reduction techniques were employed during training to prevent overfitting and optimize model performance.

- ✓ **Transfer Learning with ResNet50:**
  To enhance diabetic retinopathy detection, a ResNet50 model was constructed using a combination of pre-trained weights from the ImageNet dataset and custom layers. The model's architecture involved global average pooling and dropout layers, followed by dense layers with ReLU activation. The final output layer employed softmax activation for multi-class classification. Initially, most layers were set to be non-trainable, except for the last five layers, to focus on fine-tuning specific features. Early stopping and learning rate reduction techniques were implemented to combat overfitting. The model was compiled with the Adam optimizer, binary cross-entropy loss, and accuracy as the evaluation metric. The resulting model configuration and architecture provided a foundation for improved diabetic retinopathy detection by leveraging the power of deep learning and transfer learning techniques.

2.4 Results: -

The test accuracies of the machine models that are performed on diabetic retinopathy images are shown in table 2.1.

Models	Test Accuracy
Naïve Bayes	41.35
K Nearest neighbors	46.79
Support Vector Machines	48.12
Decision Trees	43.20
Random Forest	55.12
Gradient Boosting	55.00
XGBoost	56.28
CatBoost	56.85
AdaBoost	45.92

Table 2.1

The train and test accuracies of our deep learning models as shown in Table 2.2

Models	Train Accuracy	Test Accuracy
Artificial Dense Network	43.5%	41.7%
Convolutional Neural Network	60.03%	65.16%
InceptionV3	82.5%	78.3%
ResNet50	86.5%	81.9%

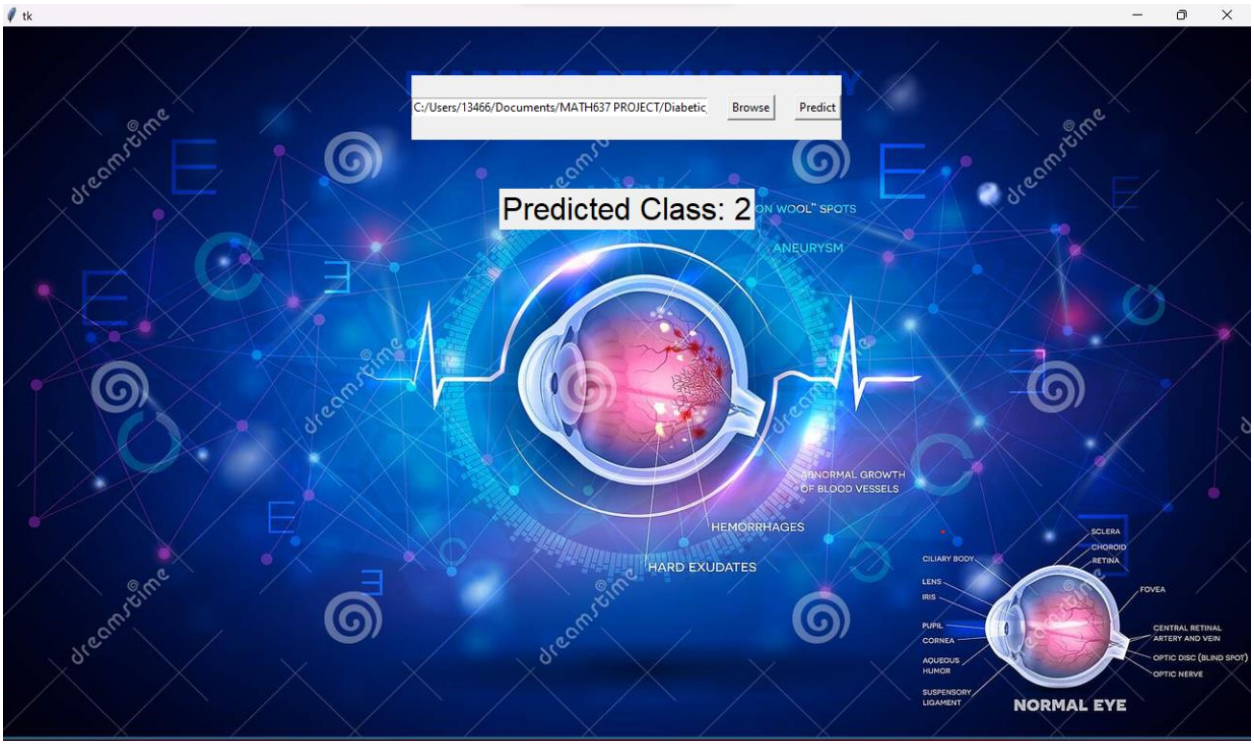
Table 2.2

After tuning the models, we got the best accuracy with ResNet50 as 81.9% test accuracy. The confusion matrix of ReNet50 is shown in fig 2.6.

2.5 Deployment:

To create an accessible platform for users to utilize the deep learning model, we developed a graphical user interface (GUI) using the Tkinter library in Python. This user-friendly interface allows individuals, even those without a technical background, to interact with the model.

The GUI design allows users to browse their local filesystem, select an image, and obtain a prediction. It accomplishes this by providing interactive buttons for users to select their image files and execute the prediction function. Behind the scenes, the selected image is automatically preprocessed and fed into the pre-trained deep learning model to generate the prediction. This smooth integration of the deep learning model into a GUI highlights the practical application of our model, and it simplifies the process of image classification for the end-users.



The above picture is a screenshot of GUI for Diabetic Retinopathy prediction where the background model that predicts is ResNet_50.

2.6 Conclusion:

The detection of diabetic retinopathy, a critical problem in preventing vision loss and blindness, has been tackled by constructing multiple models aimed at early identification and proactive measures. As the project progresses, a promising future scope lies in refining the model's parameters and exploring the utilization of advanced pretrained neural networks like EfficientNetB7. By harnessing the capabilities of these sophisticated architectures, it is anticipated that the accuracy of diabetic retinopathy detection can be significantly improved, enabling more precise and reliable identification of this condition. Such advancements hold tremendous potential for enhancing the effectiveness of preventive measures and ultimately minimizing the risk of blindness associated with diabetic retinopathy.

3.0 References:

[1] "Predicting Diabetes Mellitus with Machine Learning Techniques by Quan zou, Kyang Qu, Yamei Luo, www.frontiersin.org/articles/10.3389/fgene.2018.00515/full"

[2] "Prediction of diabetes using machine learning in health care by Muhammed Azeem. Nasir Kamal, https://ieeexplore.ieee.org/document/8748992"

[3] "Diabetic Retinopathy Detection using Deep Learning by Supriya,Seema,Zia, https://ieeexplore.ieee.org/document/9277506"

[4] "Diabetic Retinopathy detection using deep learning techniques: A Review by Wejdan L, Wafaa M. https://www.sciencedirect.com/science/article/pii/S2352914820302069"

[5] "Diabetic Prediction Dataset https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset"

[6] "Diabetic Retinopathy Dataset https://www.kaggle.com/competitions/aptos2019-blindness-detection/data"

4.0 Credits:

Diabetic Prediction and Diabetic Retinopathy detection codes are implemented on own by inspiring the notebooks that are submitted in Kaggle competitions.

GUI implementation for Diabetic Prediction is built using streamlit library learnt and inspired from YouTube videos.

Tkinter library is used in building GUI for Diabetic Retinopathy, partial codes are taken from ChatGPT.

Background Image used in DR GUI taken from the link https://thumbs.dreamstime.com/z/eye-diabetic-retinopathy-eye-diabetic-retinopathy-vision-disorder-normal-eye-anatomy-abstract-blue-scientific-174050244.jpg

Work Done by teammates:

Student Name	Abstract	Preprocessing	Model Building	Presentation	GUI	Report
Sri Vishnuvardhan Reddy Akepati	✓	✓	✓	✓	✓	✓
Sai Vardhan kataru	✓		✓	✓	✓	✓
Nagendra	✓		✓	✓	✓	
Charitha	✓	✓	✓		✓	
Sujith	✓	✓	✓	✓		

Table of Figures:

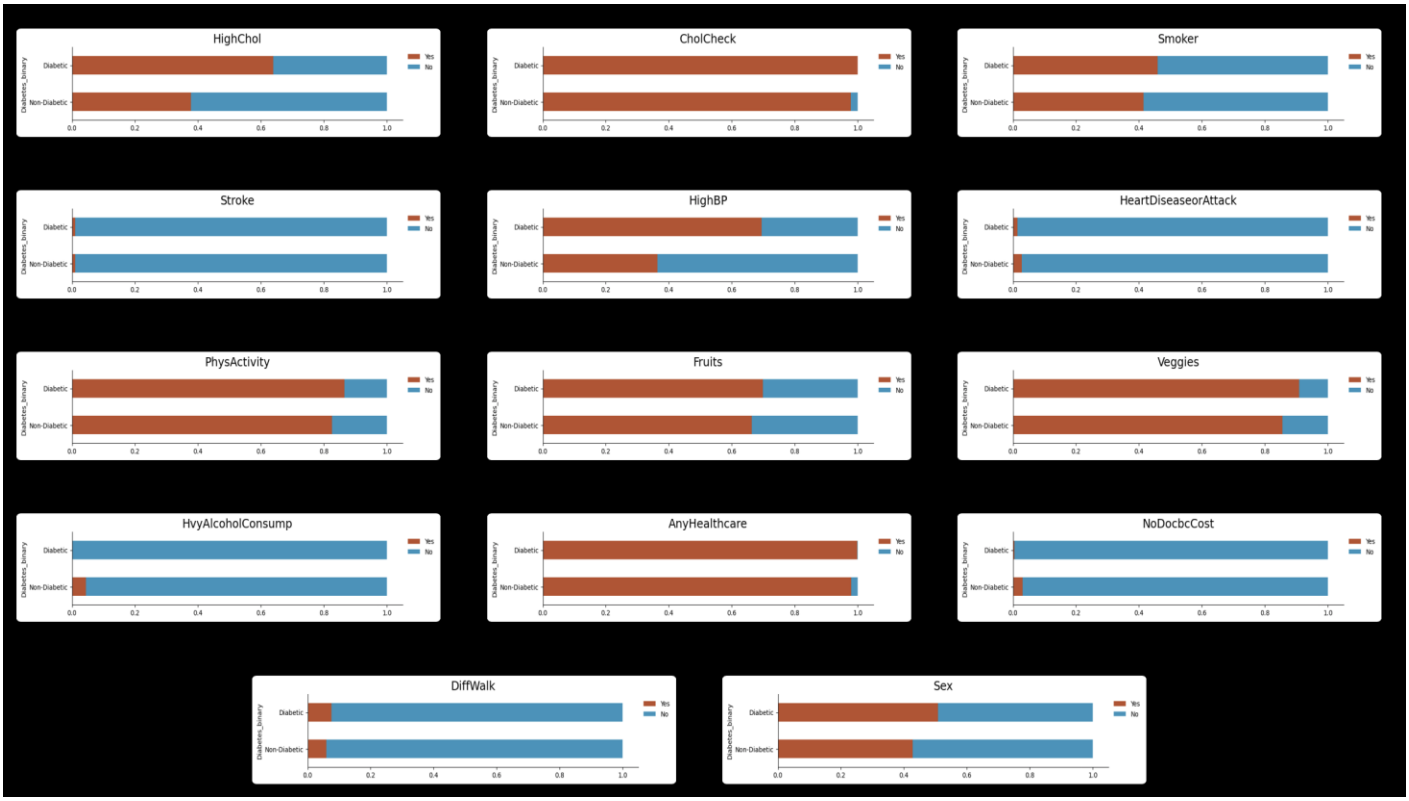

HighChol
CholCheck
Smoker
Stroke
HighBP
HeartDiseaseorAttack
PhysActivity
Fruits
Veggies
HvyAlcoholConsump
AnyHealthcare
NoDocbcCost
DiffWalk
Sex
Diabetic
Non-Diabetic
Yes
No

Fig 1.1

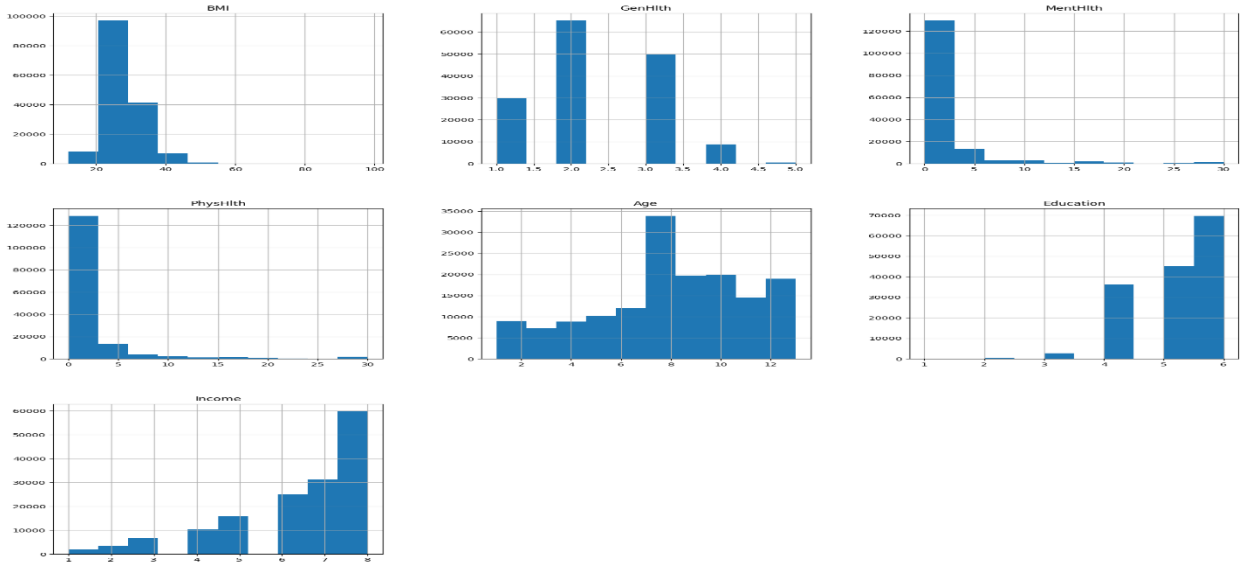

BMI
GenHlth
MentHlth
PhysHlth
Age
Education
Income

Fig 1.2

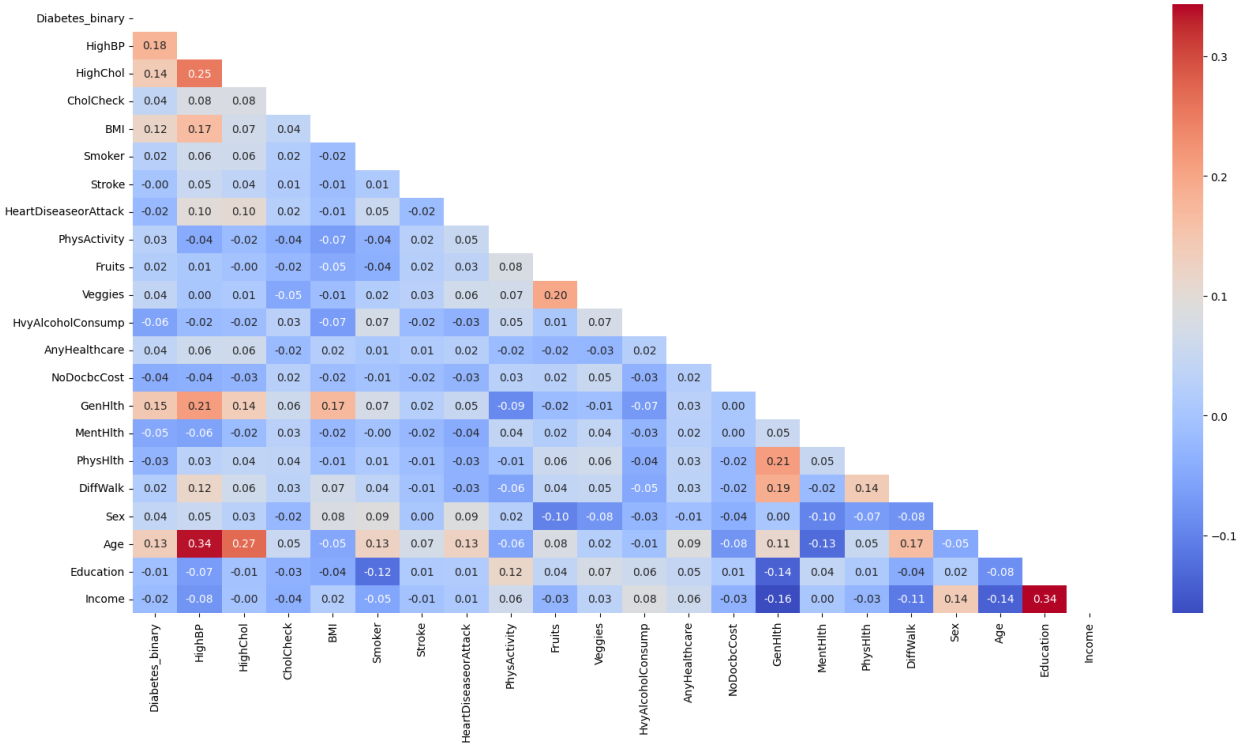


Fig 1.3

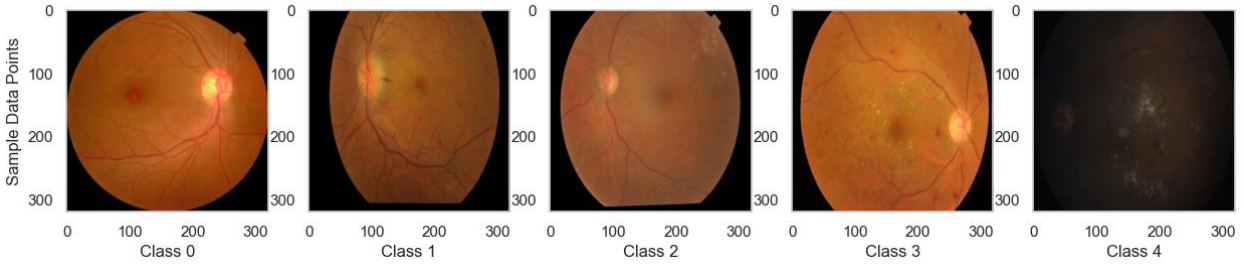


Fig 2.1

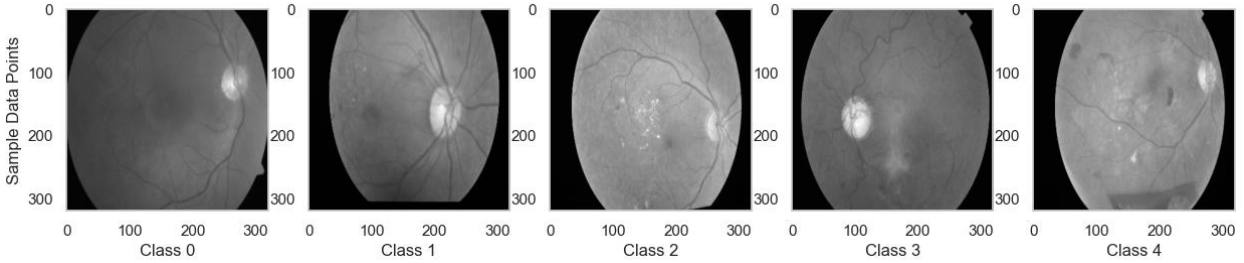


Fig 2.2

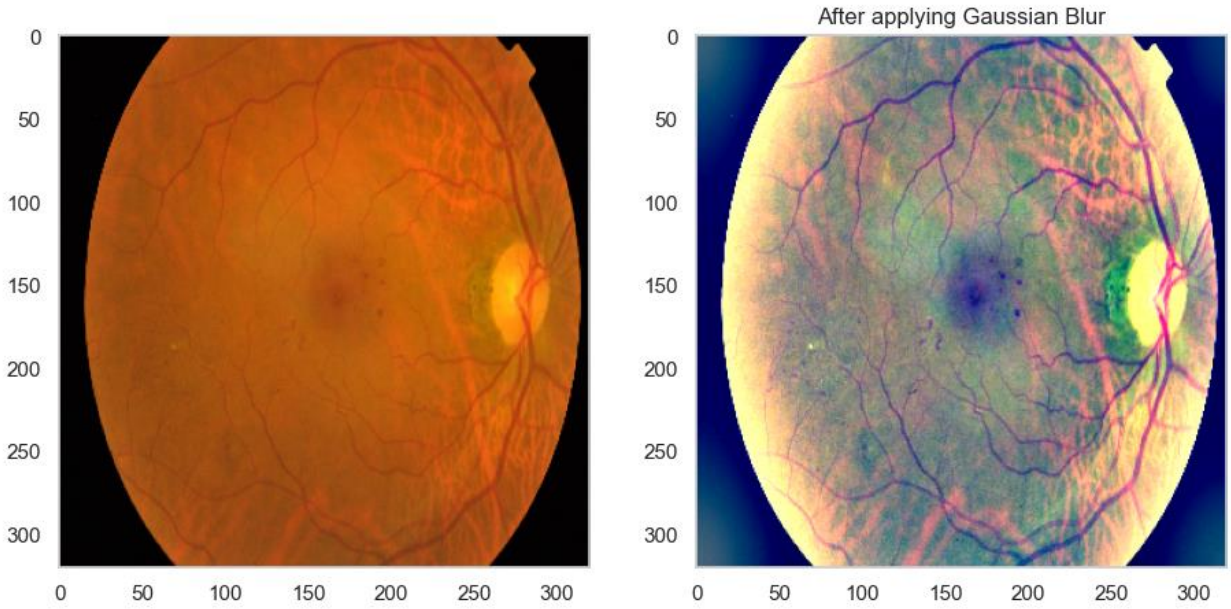

After applying Gaussian Blur
0
50
100
150
200
250
300

Fig 2.3

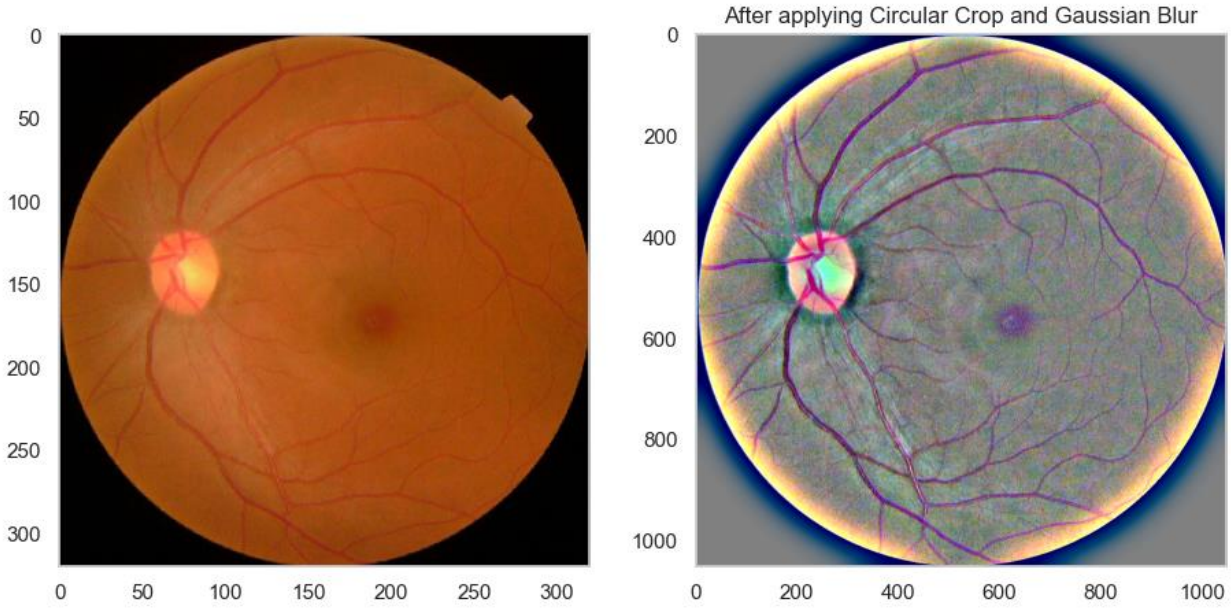

After applying Circular Crop and Gaussian Blur
0
50
100
150
200
250
300
400
600
800
1000

Fig 2.4

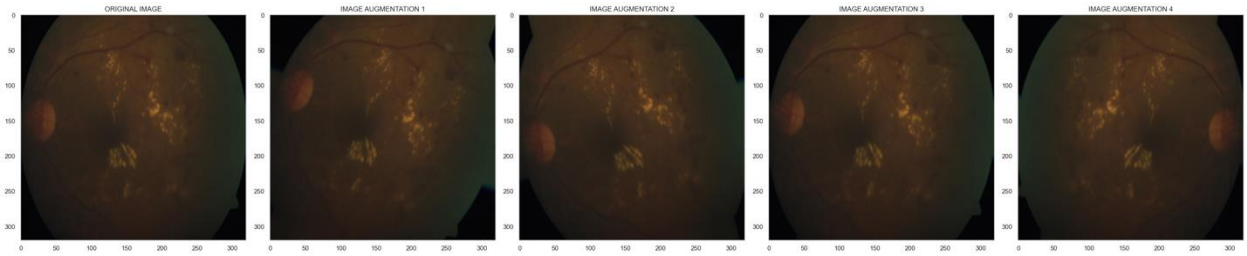

ORIGINAL IMAGE
IMAGE AUGMENTATION 1
IMAGE AUGMENTATION 2
IMAGE AUGMENTATION 3
IMAGE AUGMENTATION 4

Fig 2.5

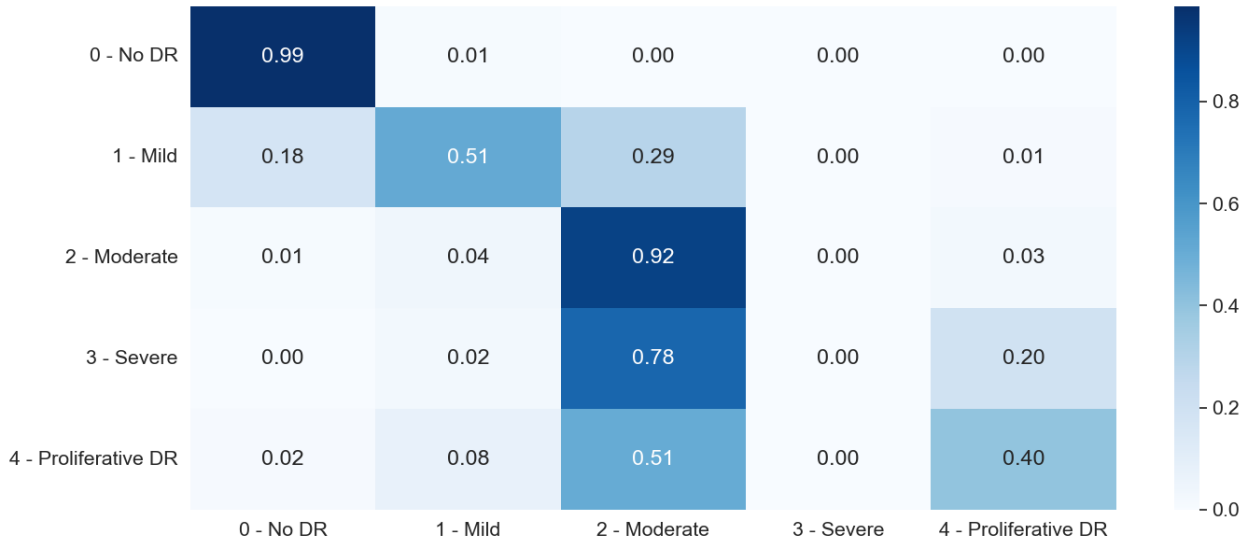

0 - No DR
1 - Mild
2 - Moderate
3 - Severe
4 - Proliferative DR
0.99 0.01 0.00 0.00 0.00
0.18 0.51 0.29 0.00 0.01
0.01 0.04 0.92 0.00 0.03
0.00 0.02 0.78 0.00 0.20
0.02 0.08 0.51 0.00 0.40
0 - No DR
1 - Mild
2 - Moderate
3 - Severe
4 - Proliferative DR
0.8
0.6
0.4
0.2
0.0

Fig 2.6