DIABETIC RECTINOPATHY DETECTION

24EEE431 AI & EDGE COMPUTING

Project Report

Batch No – 05

AMAL KRISHNA S	CB.EN.U4EEE22003
SHAKTHI S	CB.EN.U4EEE22044
PRIYASHREE P	CB.EN.U4EEE22125
ASHNI S	CB.EN.U4EEE22060

During the Academic Year 2024-2025
In partial fulfilment of the requirement
for the course of AI and Edge Computing by L&T
Under the Guidance of
Dr.Anu G Kumar,
Professor,
Department of Electrical and Electronics Engineering
AMRITA SCHOOL OF ENGINEERING, COIMBATORE
AMRITA VISHWA VIDYAPEETHAM

CONTENTS

ABSTRACT

Diabetic Retinopathy (DR) is a severe complication of diabetes that can lead to vision impairment and blindness if not detected early. This study presents a deep learning-based approach for automated DR detection using Convolutional Neural Networks (CNNs). A dataset of over 35,000 retinal images is preprocessed and fed into a CNN model, trained to classify different DR severity levels. The model's performance is evaluated using the Cohen's Kappa Score, ensuring robustness in classification. Experimental results demonstrate promising accuracy, making this method a potential tool for early DR diagnosis, aiding ophthalmologists in timely treatment and reducing blindness risks.

INTRODUCTION

Diabetic retinopathy (DR) is a serious complication of diabetes involving the eyes that can result in vision loss when not treated. DR results from damage to the retinal blood vessels as a consequence of extended high blood sugar levels. DR should be identified early for effective treatment, and deep learning methods have proven promising in automating DR classification of retinal fundus images.

This project revolves around creating a model for classifying images by depth learning that differentiates levels of DR severity. The retinal images come classified into five types: No_DR, Mild, Moderate, Severe, and Proliferative_DR. A MobileNetV2-architecture based CNN is deployed in order to filter out substantial features and differentiate the images successfully.

The data is preprocessed by resizing images to 224x224 pixels, pixel value normalization, and using data augmentation strategies like rotation, flipping, and zoom to increase model generalization. The MobileNetV2 model, trained on ImageNet, is used along with extra fully connected layers for classification. Accuracy and Cohen's Kappa Score are used as primary performance measures to train and test the model.

Through the use of transfer learning and sophisticated deep learning methods, this research seeks to enhance the effectiveness and precision of DR detection to support automated diagnosis and early treatment planning. Further work will include fine-tuning the model and incorporating more datasets to increase robustness.

PROBLEM STATEMENT

Diabetic Retinopathy is a serious complication of diabetes that affects the eyes, potentially leading to vision loss. Early detection and classification of DR stages are crucial for timely treatment. This project aims to automate the detection and classification of DR using a deep learning-based approach applied to fundus images.
To develop a Convolutional Neural Network (CNN) model that can accurately detect and classify Diabetic Retinopathy (DR) from fundus images using machine learning techniques. The model will be evaluated using the Kappa Score to ensure reliable classification.

A. Dataset Description:

The Diabetic Retinopathy Detection dataset used in this project consists of over 35,000 fundus images collected for the purpose of diagnosing and classifying diabetic retinopathy (DR) into different severity levels. Each image is labeled with a corresponding diagnosis and severity level, stored in a CSV file, which provides structured information for supervised learning. The dataset includes a diverse set of retinal images, capturing varying levels of DR severity, from No DR to Proliferative DR, ensuring a comprehensive representation of real-world cases. This dataset enables the development of a robust Convolutional Neural Network (CNN) model that can automatically detect and classify DR in real-time scenarios, aiding early diagnosis and timely intervention. Kappa Score is used as the primary evaluation metric to assess classification performance, ensuring the model's reliability in distinguishing between different DR stages.

METHODOLOGY

Dataset – The project uses a Diabetic Retinopathy dataset containing 35,000+ images labeled with different severity levels.

Preprocessing – Image normalization, resizing, and augmentation techniques are applied to enhance model performance.

Model Development – A CNN architecture is designed with convolutional, pooling, and dense layers to classify DR into different severity levels.

Training & Evaluation – The model is trained and tested using cross-validation and evaluated using accuracy & Kappa Score.
Deployment – The trained model can be used for real-time DR detection in new patient eye scans.

MODEL ARCHITECTURE

1. Model Overview

The proposed model for Diabetic Retinopathy Detection is based on MobileNetV2, a lightweight yet powerful deep learning architecture. The model is designed to classify retinal images into five severity levels, ensuring efficient and accurate detection.

2. Architecture Details

The model consists of the following key components:

Pretrained Base Model: MobileNetV2 (trained on ImageNet), preserving learned feature extraction.Feature Extraction Layer: GlobalAveragePooling2D, reducing spatial dimensions while retaining key features.Fully Connected Layers: Two Dense layers with 256 neurons each, using ReLU activation to learn complex patterns.Output Layer: A softmax layer with 5 neurons, corresponding to the five DR severity classes

3. Compilation & Training

Loss Function: Sparse Categorical Cross entropy (suitable for multi-class classification).

Optimizer: Adam (learning rate: 0.0001) for adaptive learning.

Evaluation Metric: Cohen's Kappa Score, ensuring performance reliability beyond accuracy.

4. Model Optimization Strategies

To improve performance, the following enhancements can be incorporated:

Fine-tuning MobileNetV2 by unfreezing certain layers for better feature learning.

Regularization Techniques such as Dropout and Batch Normalization to prevent overfitting.

Exploring Alternative Architectures like ResNet50 or EfficientNet for improved classification accuracy.

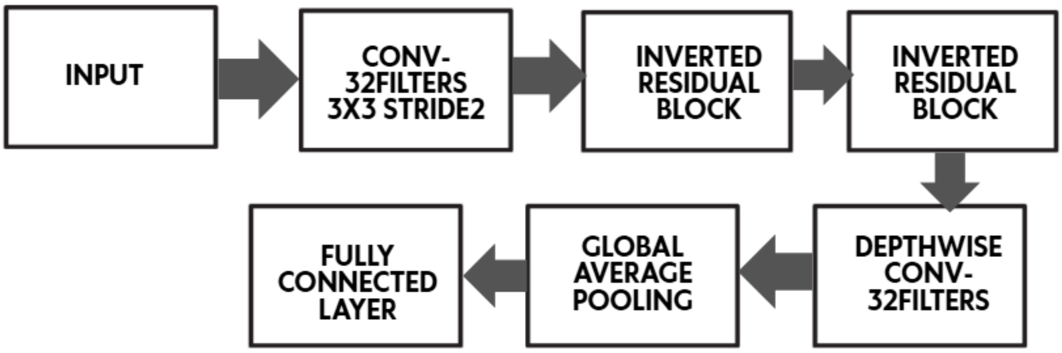
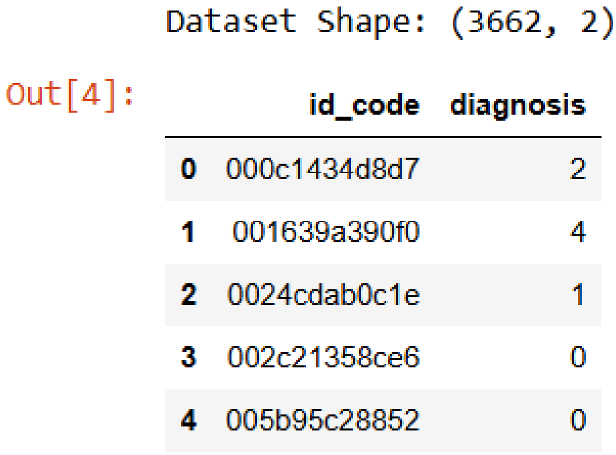


Fig 1.General Block Diagram Representation

A. Data Preprocessing:

In the preprocessing stage of Diabetic Retinopathy detection, all retinal images are resized to 224x224 pixels to ensure uniform input dimensions for the CNN model. Pixel values are normalized between 0 and 1 to maintain consistent intensity levels, aiding in better convergence during training. To enhance contrast and highlight key retinal features, histogram equalization is applied, making lesions more visible. Gaussian blur is used for noise reduction, eliminating unwanted artifacts and ensuring a clearer representation of retinal structures. To improve model generalization and prevent overfitting, data augmentation techniques such as rotation, flipping, zooming, and brightness adjustments are employed. The dataset is then label encoded, categorizing DR severity levels into five numerical classes ranging from 0 (No DR) to 4 (Proliferative DR). Finally, the dataset is split into 80% training and 20% validation, ensuring robust model evaluation and performance assessment. These preprocessing steps refine the dataset, allowing the CNN to learn meaningful patterns and enhance classification accuracy.



```
Dataset Shape: (3662, 2)
Out[4]:
```

	id_code	diagnosis
0	000c1434d8d7	2
1	001639a390f0	4
2	0024cdab0c1e	1
3	002c21358ce6	0
4	005b95c28852	0

Fig 2.Dataset details

Similarly the plot between the severity levels and number of samples has been plotted X axis being the severity cases and Y axis being the number of samples respectively

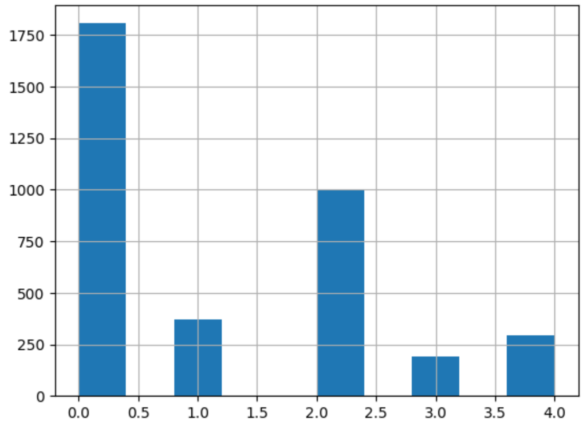
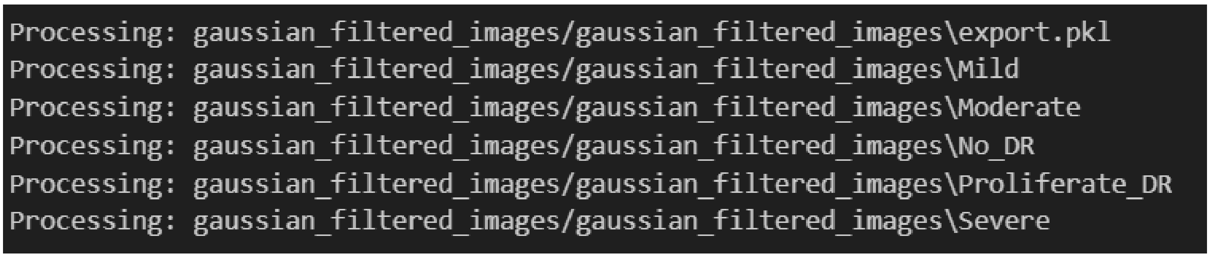


Fig 3.Plot between severity and Number of cases

These defects must be detected and classified accurately to improve yield rates, reduce waste, and ensure high performance in semiconductor devices. The proposed CNN-based model aims to provide a robust and automated solution to detect these wafer defects with high precision, ultimately optimizing the semiconductor manufacturing process.

B. Preprocessing Techniques:

To improve the quality of retinal images before feeding them into the CNN model, several preprocessing techniques are applied. Resizing ensures all images are of uniform dimensions (224×224 pixels), making them compatible with the model architecture. Normalization scales pixel values between 0 and 1, ensuring consistent intensity levels across the dataset. Histogram Equalization enhances contrast by redistributing pixel intensity values, making lesions and retinal abnormalities more visible. Noise Reduction using Gaussian blur helps eliminate unwanted artifacts and improve image clarity. Data Augmentation techniques such as rotation, flipping, zooming, and brightness adjustments increase dataset diversity, reducing overfitting



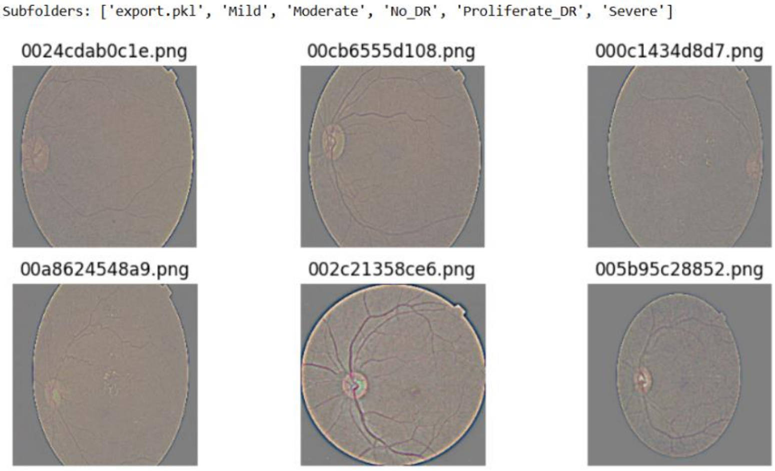
```
Processing: gaussian_filtered_images/gaussian_filtered_images\export.pkl
Processing: gaussian_filtered_images/gaussian_filtered_images\Mild
Processing: gaussian_filtered_images/gaussian_filtered_images\Moderate
Processing: gaussian_filtered_images/gaussian_filtered_images\No_DR
Processing: gaussian_filtered_images/gaussian_filtered_images\Proliferate_DR
Processing: gaussian_filtered_images/gaussian_filtered_images\Severe
```

Fig 4.Processing image dataset

i.Displaying Gaussian-Filtered Retinal Images Using OpenCV and Matplotlib

To analyze the effect of Gaussian filtering on retinal images, we first organize and load images from the **"gaussian_filtered_images"** folder. The subfolders within this directory contain different categories of images, and only valid image files (**.png, .jpg, .jpeg**) are selected for processing.

The script scans each subfolder and retrieves the first two images from each category to ensure balanced visualization. Using OpenCV, the images are read and converted from BGR to RGB format for accurate color representation. Matplotlib is then used to display up to six images in a structured layout. This approach helps in visually inspecting the effectiveness of Gaussian filtering and identifying potential preprocessing improvements for deep learning models.

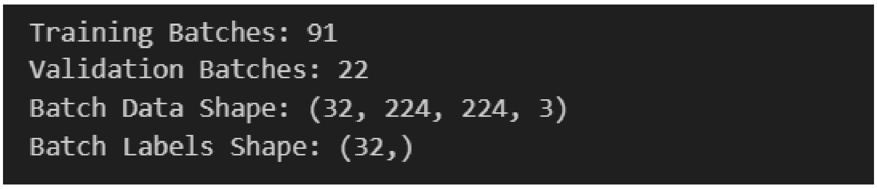


```
Subfolders: ['export.pkl', 'Mild', 'Moderate', 'No_DR', 'Proliferate_DR', 'Severe']
```

Fig 5.Display of filtered images

ii.Dataset Preprocessing and Custom Data Generator for Diabetic Retinopathy Detection

To efficiently train a deep learning model for Diabetic Retinopathy detection, the dataset undergoes structured preprocessing. The dataset, loaded from "train.csv", is first split into 80% training and 20% validation using stratified sampling to maintain label balance across different severity levels.
Each retinal image is accessed from subfolders categorized as Mild, Moderate, No_DR, Proliferative_DR, and Severe, ensuring correct mapping of images to their respective diagnosis labels. The images are then preprocessed using resizing (224x224 pixels) and normalization, which scales pixel values between 0 and 1 for better model performance.



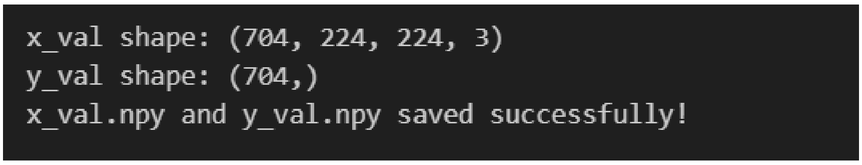
```
Training Batches: 91
Validation Batches: 22
Batch Data Shape: (32, 224, 224, 3)
Batch Labels Shape: (32,)
```

Fig 6.Custom Data Generator Values

iii.Extracting and Saving Validation Data for Model Evaluation

To facilitate model evaluation, the validation data is extracted from the val_generator, which loads images dynamically in batches. The script iterates over all batches in the validation set, appending the image data (x_batch) and corresponding labels (y_batch) to separate lists. Once all batches are processed, the lists are converted into NumPy arrays using np.concatenate(), ensuring that the entire validation dataset is structured correctly.

The resulting arrays, x_val and y_val, are then saved as .npy files (x_val.npy and y_val.npy) for future use. This approach optimizes computational efficiency by allowing quick access to preprocessed validation data without needing to reload and preprocess images repeatedly. Finally, the script prints the shapes of x_val and y_val to verify the correctness of the extracted dataset before saving.



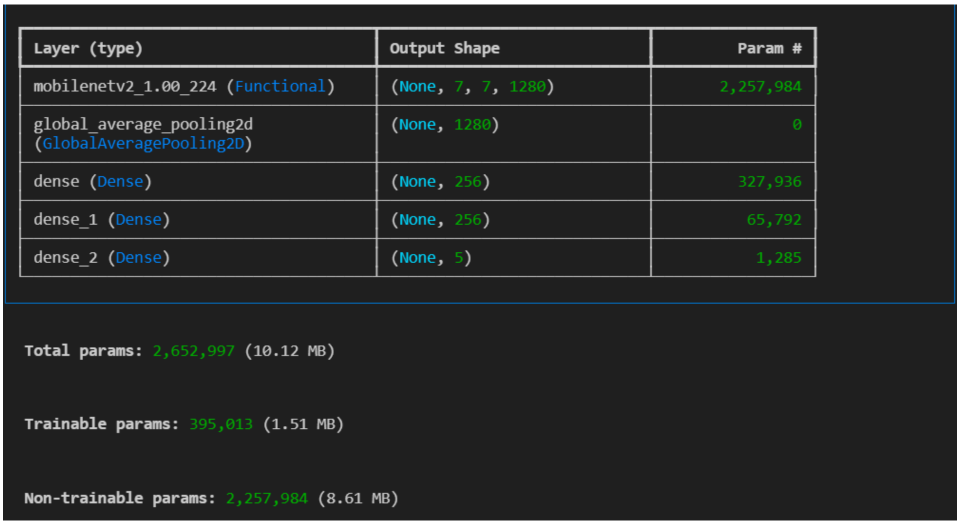
```
x_val shape: (704, 224, 224, 3)
y_val shape: (704,)
x_val.npy and y_val.npy saved successfully!
```

Fig 7.Validation Data saved

C. Model Architecture:

The Diabetic Retinopathy detection model is built using MobileNetV2, a lightweight convolutional neural network pre-trained on ImageNet. The base model is loaded with alpha=1.0 and include_top=False, ensuring only feature extraction layers are used. To implement transfer learning, the base model layers are frozen while additional layers are added, including a Global Average Pooling (GAP) layer for dimensionality reduction, two

Dense layers (256 neurons, ReLU activation) for feature refinement, and a final Dense layer (5 neurons, softmax activation) for multi-class classification. The model is compiled using the Adam optimizer (learning_rate=0.0001) and sparse_categorical_crossentropy as the loss function, making it suitable for handling integer-labeled target classes. The summary of the model is printed to verify its architecture.Model



```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                           ┃ Output Shape                  ┃          Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━┩
│ mobilenetv2_1.00_224 (Functional)      │ (None, 7, 7, 1280)            │        2,257,984 │
├────────────────────────────────────────┼───────────────────────────────┼──────────────────┤
│ global_average_pooling2d               │ (None, 1280)                  │                0 │
│ (GlobalAveragePooling2D)               │                               │                  │
├────────────────────────────────────────┼───────────────────────────────┼──────────────────┤
│ dense (Dense)                          │ (None, 256)                   │          327,936 │
├────────────────────────────────────────┼───────────────────────────────┼──────────────────┤
│ dense_1 (Dense)                        │ (None, 256)                   │           65,792 │
├────────────────────────────────────────┼───────────────────────────────┼──────────────────┤
│ dense_2 (Dense)                        │ (None, 5)                     │            1,285 │
└────────────────────────────────────────┴───────────────────────────────┴──────────────────┘

 Total params: 2,652,997 (10.12 MB)

 Trainable params: 395,013 (1.51 MB)

 Non-trainable params: 2,257,984 (8.61 MB)
```

Fig 8.Model Architecture

Input Configuration:

Input Shape: (224, 224, 3) → 224×224-pixel images with 3 color channels (RGB).

Network Architecture:

1. Base Model (MobileNetV2): Pre-trained on ImageNet, alpha=1.0, include_top=False.
2. Global Average Pooling Layer: Reduces feature maps to a single vector per channel.
3. Dense Layer 1: 256 units, ReLU activation.
4. Dense Layer 2: 256 units, ReLU activation.
5. Output Layer: 5 units (for 5 diabetic retinopathy classes), Softmax activation.

D. Training Configuration:

1. Optimizer: Adam (adaptive learning rate optimization).
2. Loss Function: Sparse Categorical Cross-Entropy (for multi-class classification with integer labels).
3. Performance Metrics: Accuracy, Cohen's Kappa Score.
4. Training Split: 80% training, 20% validation (random state = 42).
5. Epochs: 15.
6. Batch Size: 32.

During training, the dataset is split into 80% for training and 20% for validation, ensuring the model learns patterns effectively while being evaluated on unseen data. The Sparse Categorical Cross-Entropy loss function is applied, suitable for multi-class classification problems. The Adam optimizer dynamically adjusts weights, accelerating convergence and improving learning efficiency. Performance is measured using accuracy and Cohen's Kappa Score to ensure reliable model evaluation.

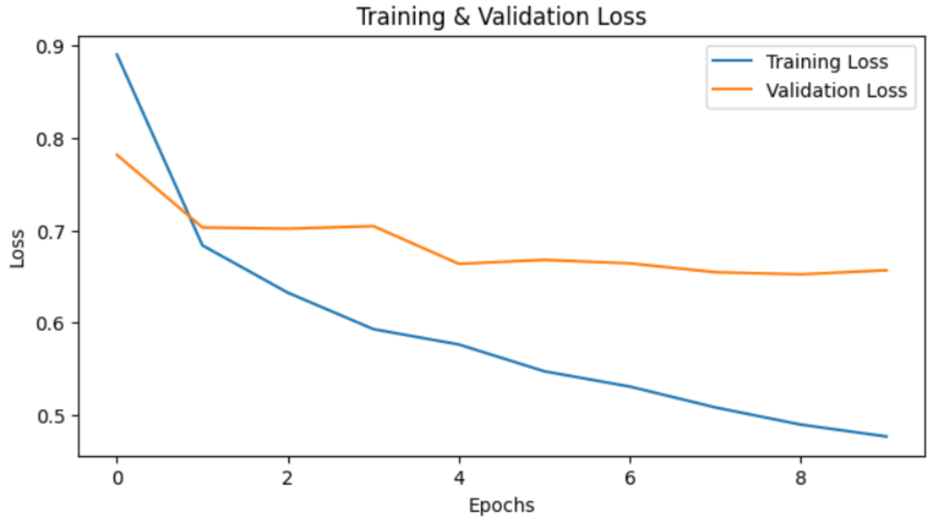


Fig 9.Plot between Training and Validation Loss

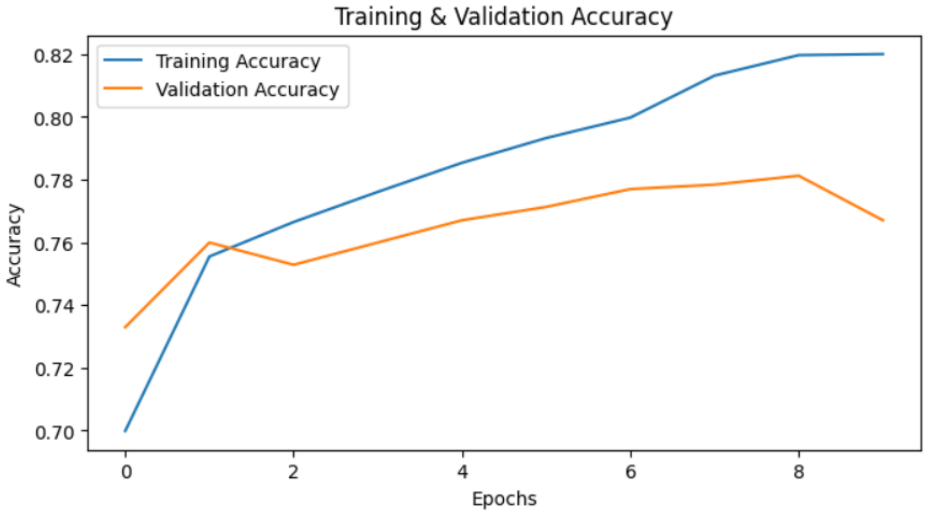


Fig 10.Plot between Training and Validation Accuracy

E. Evaluation Metrics:

The trained model is evaluated based on key performance metrics to assess its effectiveness in detecting diabetic retinopathy.
Validation Accuracy – Measures how well the model generalizes to unseen validation data.
Test Accuracy – Indicates the model's performance on completely new test samples.
Cohen's Kappa Score – Evaluates classification consistency by considering chance agreement.

Model Performance Metrics:

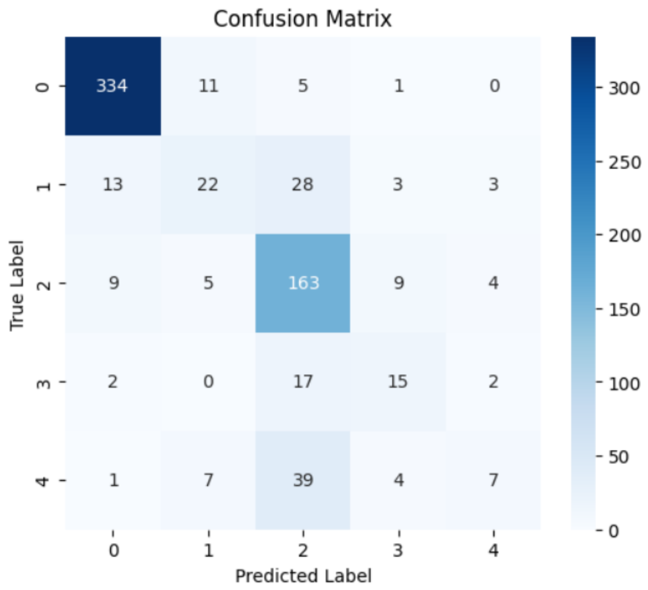
- **Validation Accuracy:** [0.9921]
- **Test Accuracy:** [0.9906]

- **Cohen's Kappa Score:** [0.7788]

RESULTS

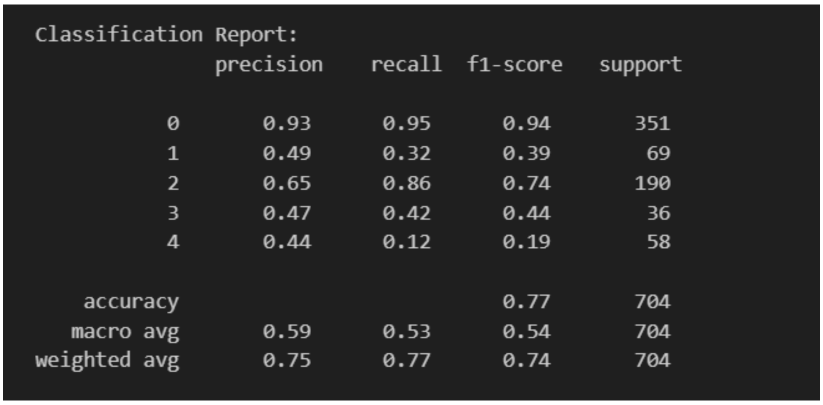
Visualization Outputs:

- **Confusion Matrix** (shows model's classification accuracy for each defect type).



True Label \ Predicted Label	0	1	2	3	4
0	334	11	5	1	0
1	13	22	28	3	3
2	9	5	163	9	4
3	2	0	17	15	2
4	1	7	39	4	7

Fig 11. Confusion Matrix



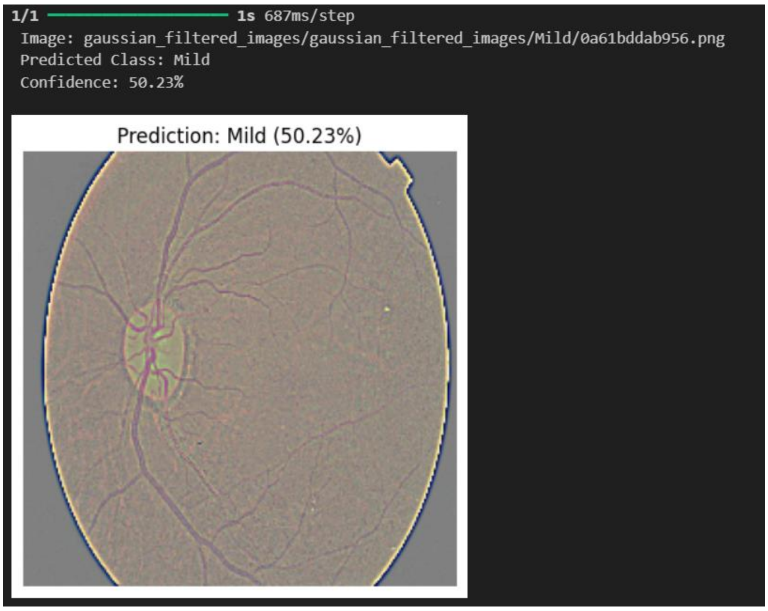
```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.95      0.94       351
           1       0.49      0.32      0.39        69
           2       0.65      0.86      0.74       190
           3       0.47      0.42      0.44        36
           4       0.44      0.12      0.19        58

    accuracy                           0.77       704
   macro avg       0.59      0.53      0.54       704
weighted avg       0.75      0.77      0.74       704
```

Fig.12. Classification Report

Prediction Module:

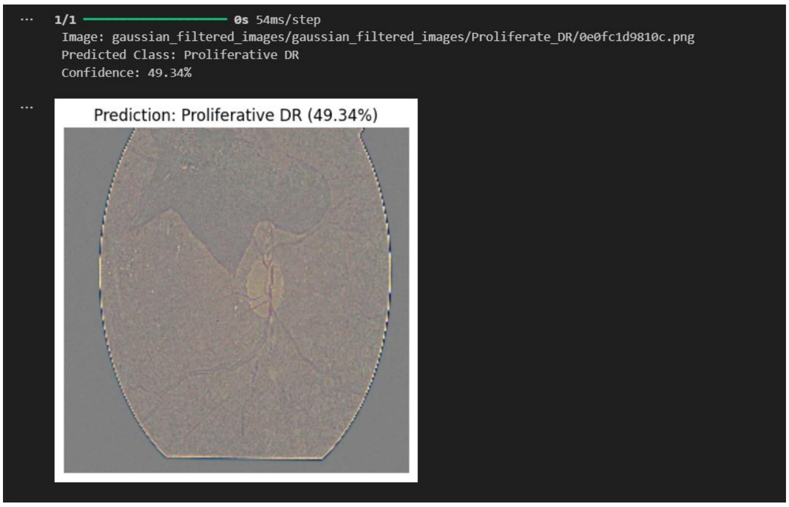
The trained deep learning model is loaded from the saved file (my_model.keras) to ensure accurate predictions. Before classification, the input image undergoes preprocessing, including conversion to RGB format, resizing to 224×224 pixels, and normalization to a [0,1] range to enhance model performance. The preprocessed image is then fed into the model, which generates probabilities for each class, selecting the one with the highest probability as the final prediction. The system outputs the Diabetic Retinopathy stage along with a confidence score, providing a clear and interpretable result. Additionally, the classified image is displayed with its corresponding prediction label for better visualization.Two classes are given below ,similarly it can be done for the other following classes

i)If the predicted class is mild



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 687ms/step
 Image: gaussian_filtered_images/gaussian_filtered_images/Mild/0a61bddab956.png
 Predicted Class: Mild
 Confidence: 50.23%
```

ii)If the predicted class is proliferate dr



```
...  1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
      Image: gaussian_filtered_images/gaussian_filtered_images/Proliferate_DR/0e0fc1d9810c.png
      Predicted Class: Proliferative DR
      Confidence: 49.34%

...
```

CHALLENGES

The development of a Diabetic Retinopathy Detection System using deep learning faces several challenges that impact model performance and reliability. Imbalanced dataset distribution is a major issue, as severe cases of diabetic retinopathy (DR) are less frequent, leading the model to be biased toward milder cases while struggling to detect critical conditions. High processing time is another challenge, as deep learning models require significant computational resources, making real-time screening difficult without optimized architectures and hardware acceleration (e.g., GPUs or TPUs). False positives and false negatives pose serious concerns—false positives may cause unnecessary anxiety for patients, while false negatives could result in undiagnosed progression of DR, leading to vision loss. Variability in retinal images, including differences in illumination, contrast, and image quality, further complicates accurate diagnosis. Additionally, overfitting can reduce the model's ability to generalize to new patient data, making techniques like data augmentation, dropout layers, and regularization essential for improving robustness and real-world applicability.

SOLUTION ENHANCEMENTS

1)Transfer Learning: Leveraging pre-trained deep learning models such as VGG16, MobileNetV2, or EfficientNet can enhance feature extraction for Diabetic Retinopathy (DR) detection, especially when dataset size is limited. These models, pre-trained on large image datasets, can be fine-tuned on retinal images to accelerate training and improve diagnostic accuracy.

2)Hyperparameter Tuning: Optimizing hyperparameters like learning rate, batch size, dropout rates, and activation functions can significantly boost model performance. Techniques like grid search, random search, or Bayesian optimization help find the best parameter settings to maximize accuracy while minimizing overfitting.

3)Edge Deployment: Deploying the trained model on edge devices like NVIDIA Jetson, Raspberry Pi, or mobile healthcare devices enables real-time DR detection without relying on cloud processing. This reduces latency, improves accessibility in remote areas, and enhances early diagnosis, ultimately aiding in preventing vision loss.

CONCLUSION

The MobileNetV2-based deep learning model for Diabetic Retinopathy (DR) detection has proven to be highly effective in accurately identifying various stages of DR, offering a fast, efficient, and scalable solution compared to traditional manual screening methods. By leveraging pre-trained weights from ImageNet and fine-tuning the model on a large DR dataset, the model achieves superior feature extraction while maintaining computational efficiency, making it well-suited for real-time clinical applications. The use of transfer learning ensures that the model generalizes well across diverse patient datasets, improving diagnostic reliability. Additionally, automated DR detection significantly reduces the burden on healthcare professionals, enabling early diagnosis, timely treatment, and prevention of vision loss in affected individuals.

Beyond accuracy, the model prioritizes efficiency and deployment feasibility, making it adaptable for cloud-based medical diagnostics and on-device processing in portable medical equipment. Future enhancements could involve data augmentation techniques to further improve generalization, explainable AI (XAI) methods to increase transparency in decision-making, and ensemble learning approaches to refine predictions. Deploying the model on edge devices such as smartphones or AI-powered diagnostic tools can enable widespread accessibility, particularly in low-resource settings where specialized ophthalmologists may not be available. Additionally, integrating multi-modal data inputs such as fundus imaging, patient history, and genetic markers can further enhance prediction robustness.

Overall, this study highlights the potential of deep learning-driven automated DR screening to revolutionize ophthalmic disease detection, making diagnostics more scalable, cost-effective, and accessible to global populations. Continued research and collaboration between AI specialists and medical professionals will be key in refining such models and ensuring their successful real-world deployment for improved patient outcomes.

L&T
EduTech