UNIVERSITY OF CALCUTTA

MCA Semester-IV, Examination - 2025
(Under CBCS)

MCA P41
Final Year Project Report

Project Report on

Integrating Explainable AI in Diabetic Retinopathy Detection

MCA 4th Sem, MCA-P41, 2023-24

Under the guidance of

Prof. Arpan Murmu
Assistant Professor, AKCSIT, University of Calcutta

University of Calcutta

Name	Registration No.	Roll Number
Arkaprabha Ray	544-1112-1220-20	C91/MCA/232003
Prantik Bhattacharya	544-1111-0667-20	C91/MCA/232012
Rahul Das	544-1112-0306-20	C91/MCA/232016

www.caluniv.ac.in

A.K. CHOUDHURY SCHOOL OF INFORMATION TECHNOLOGY
UNIVERSITY OF CALCUTTA

University of Calcutta, JD-2 Sector-III, Salt lake, Kolkata-700106, India

CERTIFICATE

This is to certify that the project synopsis entitled **"Integrating Explainable AI in Diabetic Retinopathy Detection"** submitted for partial fulfillment of the requirements of 4th Semester of Master of Computer Application (MCA),2025 under University Of Calcutta; has been carried out by :-

Arkaprabha Ray (Roll No : C91/MCA/232003 & Registration No : 544-1112-1220-20)

Rahul Das (Roll No : C91/MCA/232016 & Registration No : 544-1112-0306-20)

Prantik Bhattacharya (Roll No: C91/MCA/232012 & Registration No : 544-1111-0667-20)

under the supervision of **Prof. Arpan Murmu.**

---------- ----------

CHAIRMAN, BOARD OF EXAMINER PROJECT SUPERVISOR

External Examiner(s)

ACKNOWLEDGEMENT

We wish to express our profound sense of gratitude to our project supervisor **Prof. Arpan Murmu**, for his support, inspiration and guidance. He has showed us different ways to approach a problem. We have also learned from him that an approach needs to be persistent to accomplish our goal. We are immensely grateful to him for giving his valuable time and constant advice for discussing various ideas related to our project work. It is being precious learning experience for us to work under tutelage.

We are also thankful to our department; A.K. Choudhury School of I.T, University of Calcutta; for providing us with the required resources for working on this project.

Lastly we would like to express our heartiest gratitude to our seniors and our friends; and to all who have directly or indirectly extended their valuable guidance and advice during the preparation of this project; which gave us the continuous flow of inspiration to complete the project.

THANK YOU

Date: 10/06/2025

Arkaprabha Ray
Rahul Das
Prantik Bhattacharya

INDEX

Abstract

Diabetic retinopathy (DR) is a leading cause of vision impairment and blindness among diabetic patients. It results from prolonged high blood sugar levels causing damage to the retinal blood vessels. Early detection and timely treatment are critical for preventing significant vision loss and improving the quality of life for affected individuals. However, the manual diagnosis of DR through retinal images is labor-intensive and prone to variability among clinicians. Our project tackles these challenges by first evaluating the accuracy of various **lightweight and efficient pretrained convolutional neural networks** for classifying DR severity levels, and then implementing our own optimized model.

To enhance the interpretability of the model, we integrate **GRAD-CAM (Gradient-weighted Class Activation Mapping)**, a visualization technique that highlights the critical regions in retinal images influencing the model's predictions. This not only improves transparency but also fosters trust in automated diagnostic systems. By combining accuracy with interpretability, our approach offers a comprehensive solution to DR detection. This project report details the dataset used, preprocessing techniques, the architecture of the model, training methodology, experimental results, and future directions, showcasing the potential of this model in real-world applications, especially in resource-constrained environments.

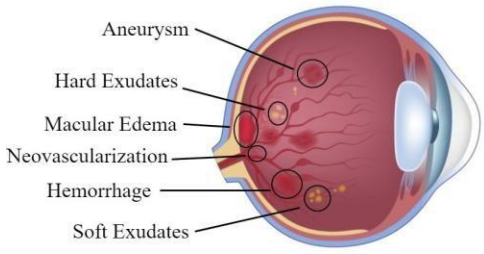


FIGURE Lesions found in patients with diabetic retinopathy.

Introduction

Diabetic retinopathy (DR) is a microvascular complication of diabetes that occurs when prolonged periods of high blood sugar cause damage to the retina's blood vessels. This condition can lead to swelling, leakage, or complete blockage of the blood vessels, ultimately affecting vision. Globally, millions of diabetic patients are at risk of DR,making it one of the leading causes of preventable blindness.

Manual detection of DR using fundus photographs is the current clinical standard but is time-consuming, requires specialized expertise, and is subject to inter-observer variability. This creates a need for automated diagnostic tools capable of identifying and classifying DR severity efficiently. Deep learning, particularly convolutional neural networks (CNNs), has emerged as a powerful approach to image analysis tasks, including medical imaging. In our project, we initially explored and compared the performance of several lightweight and efficient pretrained CNNs, ultimately implement our own optimized model employ GRAD-CAM for model interpretability, aiming to deliver a practical and trustworthy solution for DR detection.

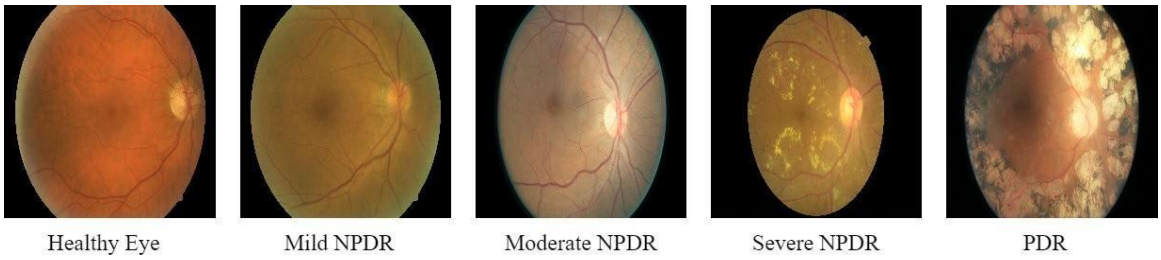


FIGURE Stages of diabetic retinopathy.

Background Study/Literature Review

Deep learning techniques have revolutionized medical image analysis, including the detection of diabetic retinopathy. CNNs, such as VGG16, ResNet, and InceptionNet have been extensively studied for their ability to identify DR features like microaneurysms, hemorrhages, and exudates. While these models achieve high accuracy, their computational demands make them less suitable for deployment in resource-constrained environments.

MobileNetV2, a lightweight neural network, addresses these limitations by using depthwise separable convolutions and linear bottlenecks to reduce the number of parameters and computation costs. Studies show thatMobileNetV2maintainscompetitiveperformancewhilebeingmore efficient. Additionally, the lack of interpretability in black-box models has been a major concern. Techniques such as GRAD-CAM provide visual explanations of model predictions by highlighting regions in the input image that influence the decision, making it possible for clinicians to validate and trust the automated systems.

Dataset:- APTOS 2019 BLINDNESS DETECTION

The APTOS dataset [13], from the Asia Pacific Tele- Ophthalmology Society (APTOS), includes high-quality retinal images and issued for the detection and grading of diabetic retinopathy. The dataset has been collected by The Aravind Eye Hospital in India. It features a diverse set of images, which are crucial for training models to handle various lighting conditions, camera models and ethnicities. The dataset consists of 3,662 images divided into five classes of which there are 1,805 fungus images for healthy eyes,370 images for mild NPDR, 999 images for moderate NPDR, 193 images for severe NPDR, and 295 images for PDR.

Transfer Learning:-

Transfer Learning (TL) is a training method aiming to adopt the weights of a pre-trained CNN and appropriately re-train the CNN to optimize the weights for a specific task i.e. AI classification of retinal image. TL is not only an efficient optimization procedure but improves classification. Early convolution layers learn to recognize generic features, edges, patterns or textures although deeper layers focus on specific aspects of the new considered image task, such as tumor segmentation or blood vessels. Actually TL is widely used in computer vision in terms of efficiency gains and reducing a major problem in the Artificial Intelligence (AI) field represented by overfitting.

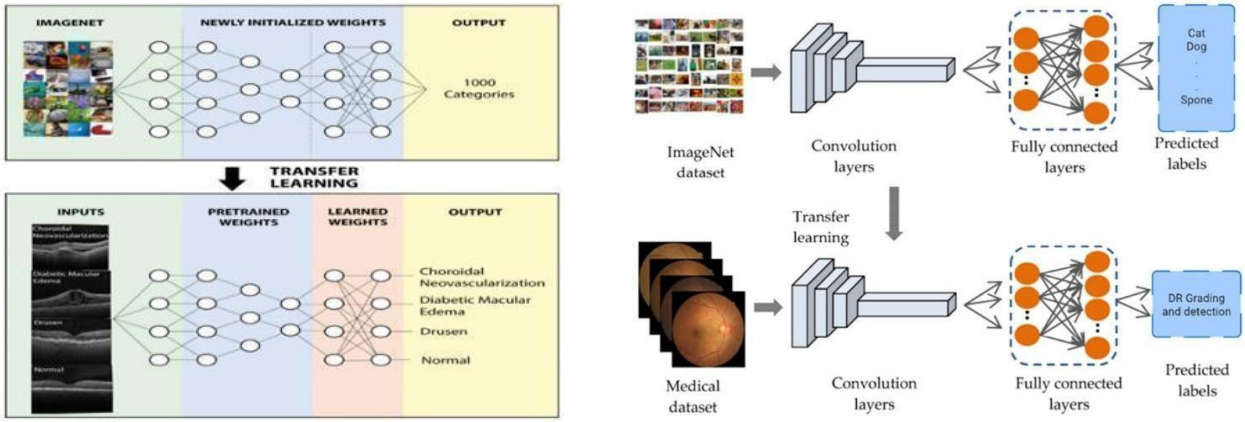
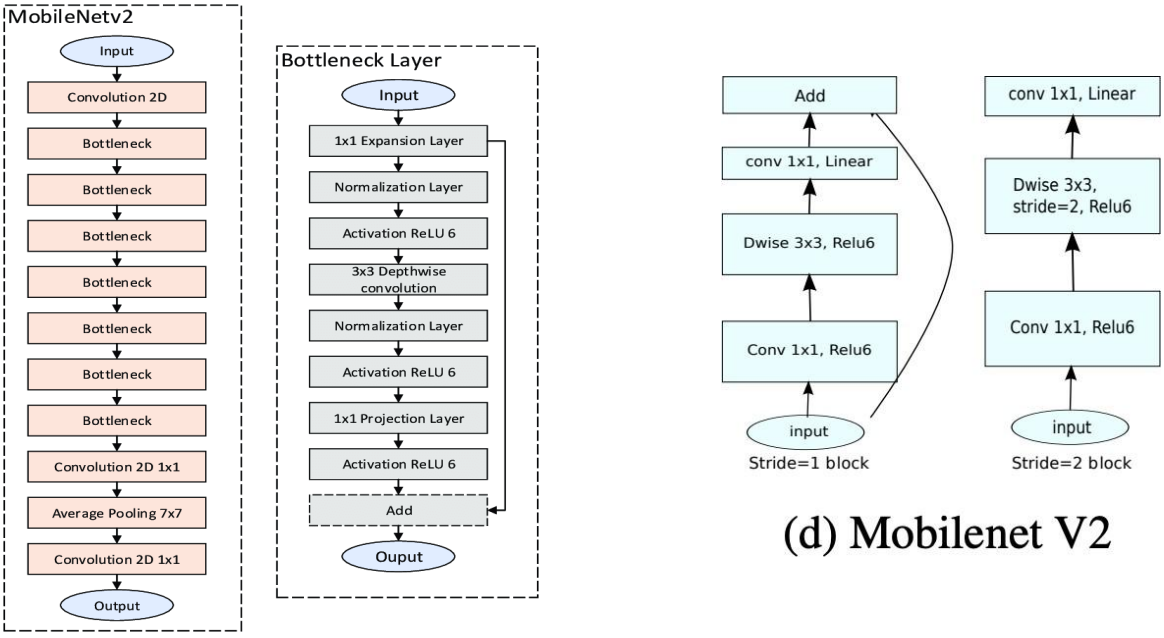


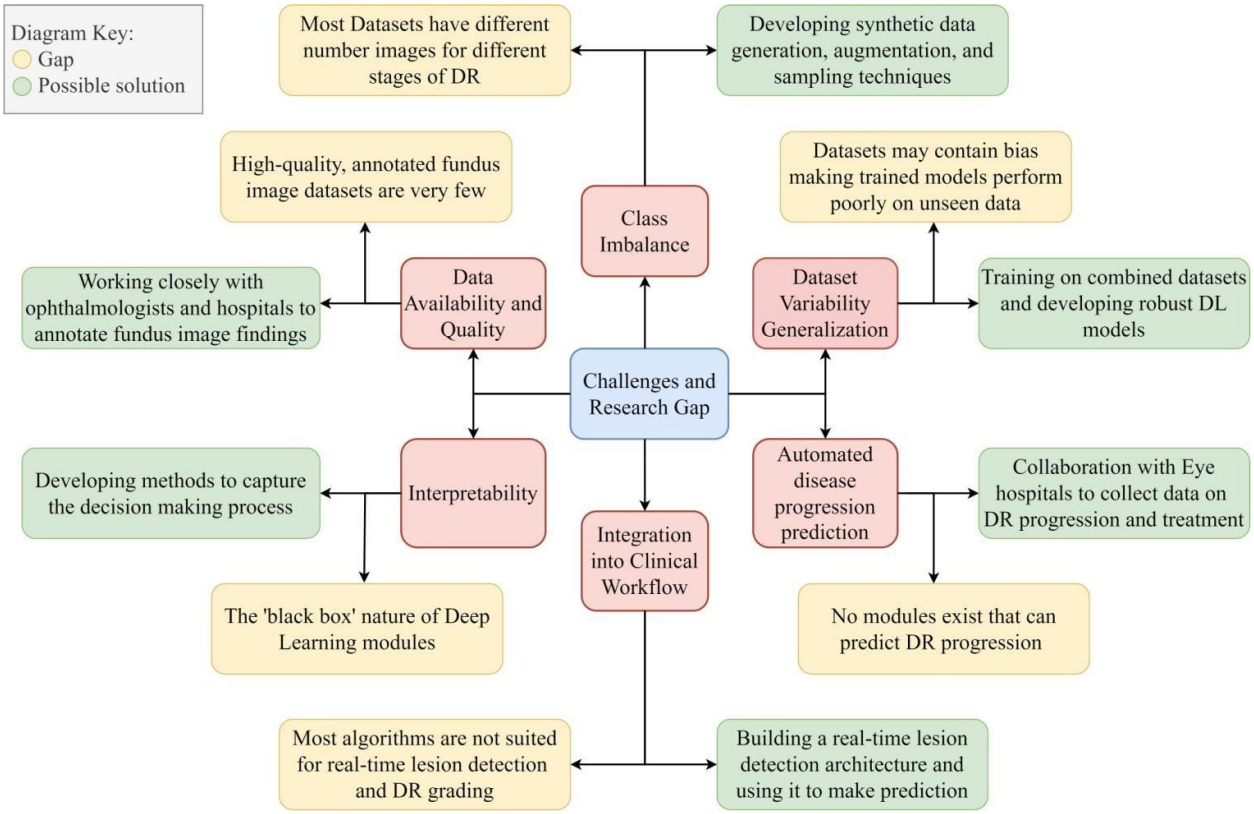
FIGURE Transfer learning in DR diagnosis.

MobileNetV2:- **MobileNetV2** is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity.As a whole, the architecture of MobileNetV2 contains the initial
Fully convolution layer with 32 filters,followed by 19 residual bottleneck layers.

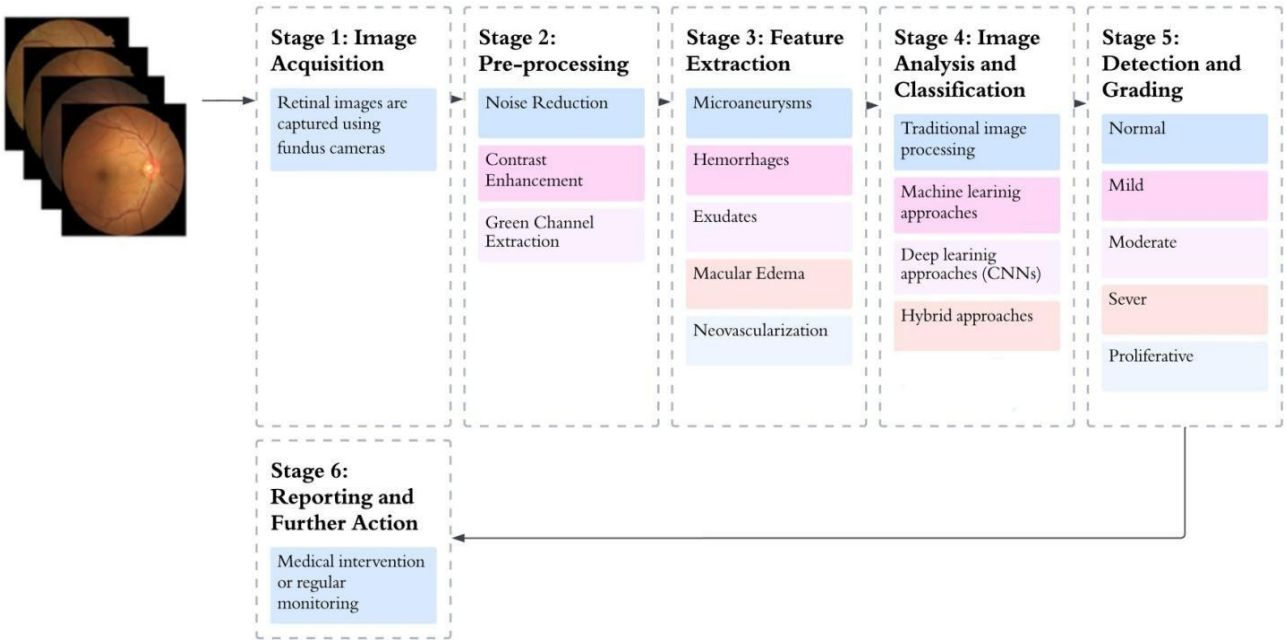


(d) Mobilenet V2

Challenges and possible solutions in the field of DR diagnosis



General Stages of detection and grading of Diabetic Retinopathy



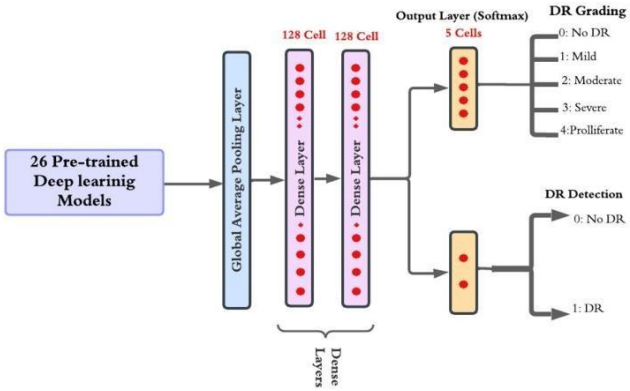


FIGURE The general stages to detect and grade DR.

Methodology

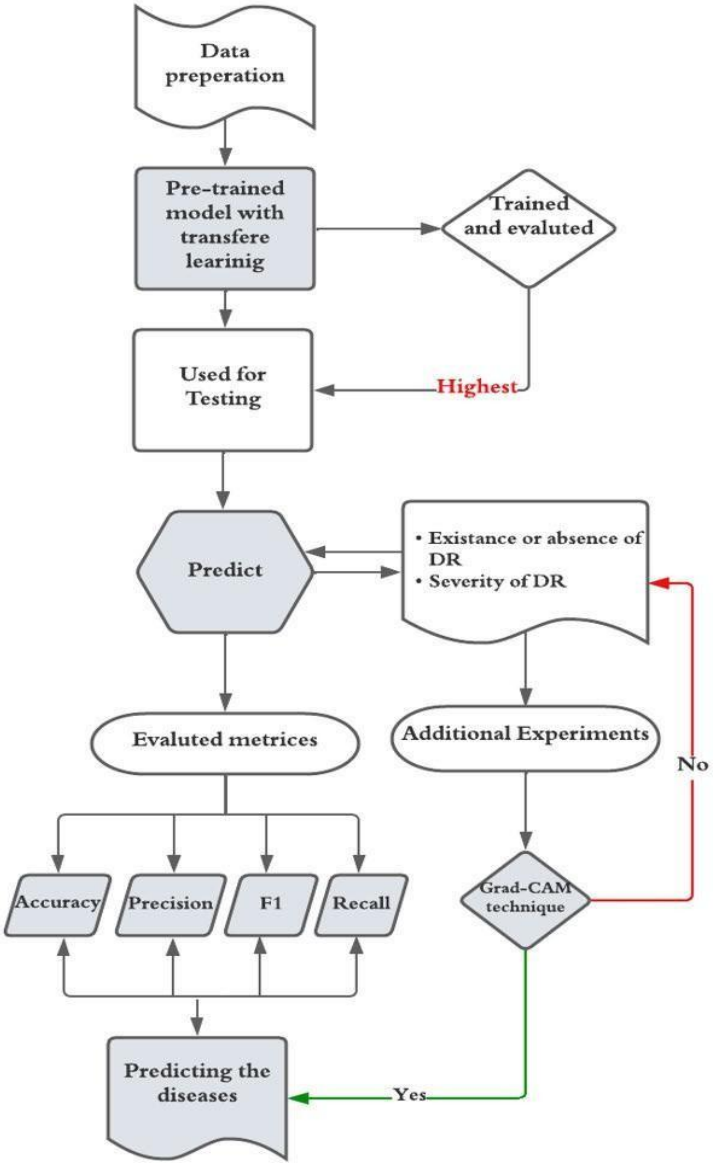
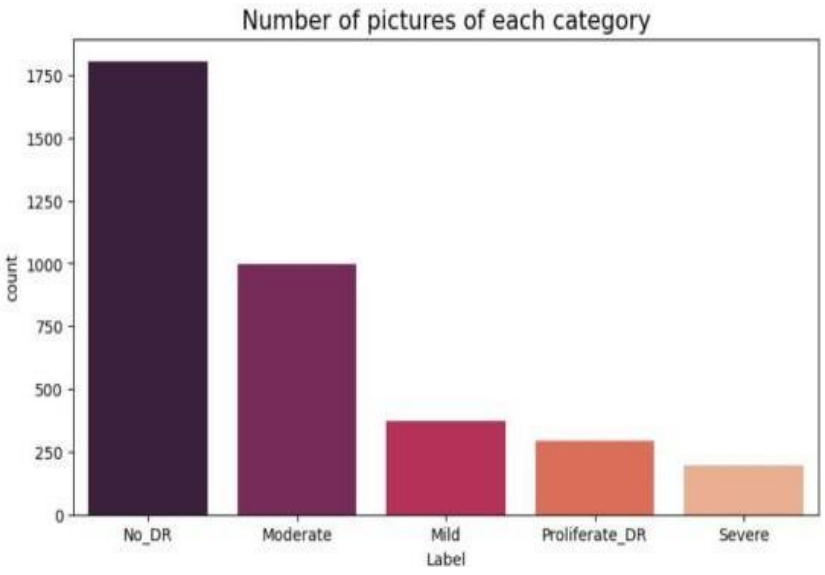


FIGURE Block diagram of the proposed methodology.

1. Dataset Preparation:

- **Dataset Source:** The Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection (APTOS 2019 BD) dataset contains 3662 samples collected from many participants of rural India. The, Aravind Eye Hospital, India, organized the dataset. The fundus photographs were collected in varying conditions and environments over a long period. Later, a group of trained doctors reviewed and labeled the gathered samples following the principle of the International Clinical Diabetic Retinopathy Disease Severity Scale (ICDRSS). As per the scaling system, the APTOS 2019 BD samples are divided into five categories:-

    - 0- No_DR
    - 1-Mild
    - 2-Moderate
    - 3-Severe
    - 4-Proliferate_DR



- **Preprocessing:**
  - Gaussian filtering retina scan images and resizing images to 224x224 pixels for consistency with various pretrained CNNs.
  - Normalizing pixel values to standardize input.
  - Data augmentation techniques such as rotation,flipping, and zooming to increase dataset diversity and prevent overfitting.

2. Model Architecture:

- **Base Model: MobileNetV2** initialized with pretrained weights from ImageNet, without the final classification layer.
- **Custom Head: Two dense layer(1024 neurons each with 'ReLU' activation function)** for feature aggregation, followed by **a softmax layer of 5neurons** to classify the images into multiple severity levels (e.g., No DR, Mild, Moderate, Severe, Proliferative).

3. Training Process:

- **Loss Function:** Categorical cross-entropy to handle multi-class classification.
- **Optimizer:** Adam optimizer with a learning rate scheduler for efficient convergence.
- **Validation:** Early stopping based on validation loss taking 'patience=2' to avoid overfitting.
- **Hardware: Kaggle's cloud environment,utilizing an NVIDIA Tesla T4 GPU**.

4. GRAD-CAM Implementation:

- GRAD-CAM generates heatmaps to visualize the critical areas in retinal images influencing the predictions.
- The heatmaps are overlaid on the original images,providing a clear interpretation of the model's focus.

Experiment

1. Experimental Setup:

- **Frameworks:** Tensor Flow and Keras were used for model development and training.
- **Hardware:** The experiments were conducted on Kaggle's cloud environment, utilizing an NVIDIA Tesla T4 GPU. This setup provided the necessary computational resources to efficiently train and evaluate the deep learning model.

2. Metrics:

- Accuracy: Measures the overall correctness of the predictions.
- Precision: Evaluates the model's ability to identify true positive cases.
- Recall (Sensitivity): Measures the model's ability to detect all relevant cases.
- F1-Score: Harmonic mean of precision and recall.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{FP+TP} \quad (2)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

3. Comparative Analysis:

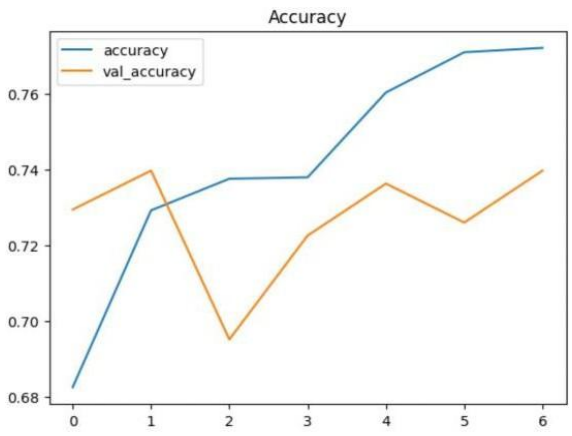
- The performance of MobileNetV2 was compared with other popular models such as ResNet50 and EfficientNet to establish its efficiency and accuracy.
- GRAD-CAM visualizations were analyzed to assess the Interpretability and clinical relevance of the model's predictions.

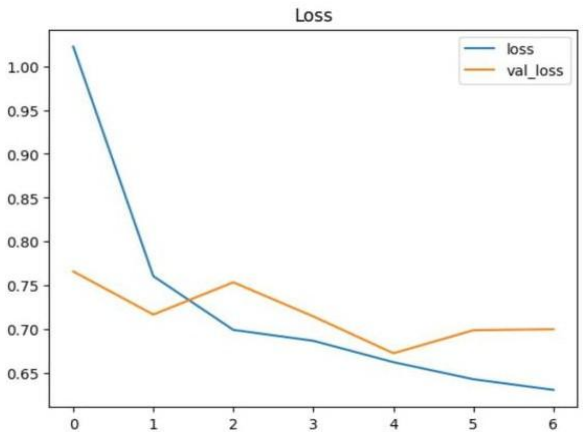
Results/Figures/Discussion

- **Comparative Analysis:**

	model	train_accuracy	val_accuracy	Training time (sec)
0	MobileNetV2	0.7242	0.7720	16.75
1	DenseNet169	0.7229	0.7660	25.34
2	ResNet152V2	0.7205	0.7599	25.97
3	InceptionV3	0.7057	0.7538	19.17
4	DenseNet201	0.7326	0.7538	27.69
5	NASNetMobile	0.7040	0.7508	29.25
6	MobileNet	0.7316	0.7508	14.66
7	InceptionResNetV2	0.6949	0.7477	28.43
8	DenseNet121	0.7161	0.7447	41.24
9	ResNet50V2	0.7245	0.7416	17.16
10	Xception	0.7097	0.7386	18.19
11	ResNet101V2	0.7222	0.7204	21.22
12	VGG16	0.6719	0.7112	15.75
13	VGG19	0.6521	0.7021	15.04
14	ResNet50	0.6524	0.6687	17.24
15	ResNet152	0.6372	0.6626	28.66
16	ResNet101	0.6291	0.6353	22.20
17	MobileNetV3Large	0.5509	0.6049	18.08
18	MobileNetV3Small	0.4727	0.4833	17.25
19	EfficientNetB1	0.4784	0.4802	22.41
20	EfficientNetB6	0.4798	0.4802	27.34
21	EfficientNetB0	0.4835	0.4802	20.60
22	EfficientNetB2	0.4710	0.4802	23.08
23	EfficientNetB3	0.4791	0.4802	25.13
24	EfficientNetB4	0.4804	0.4802	27.15
25	EfficientNetB7	0.4788	0.4802	28.17
26	EfficientNetB5	0.4761	0.4802	27.78

- **Training & Validation (MobileNetV2):**



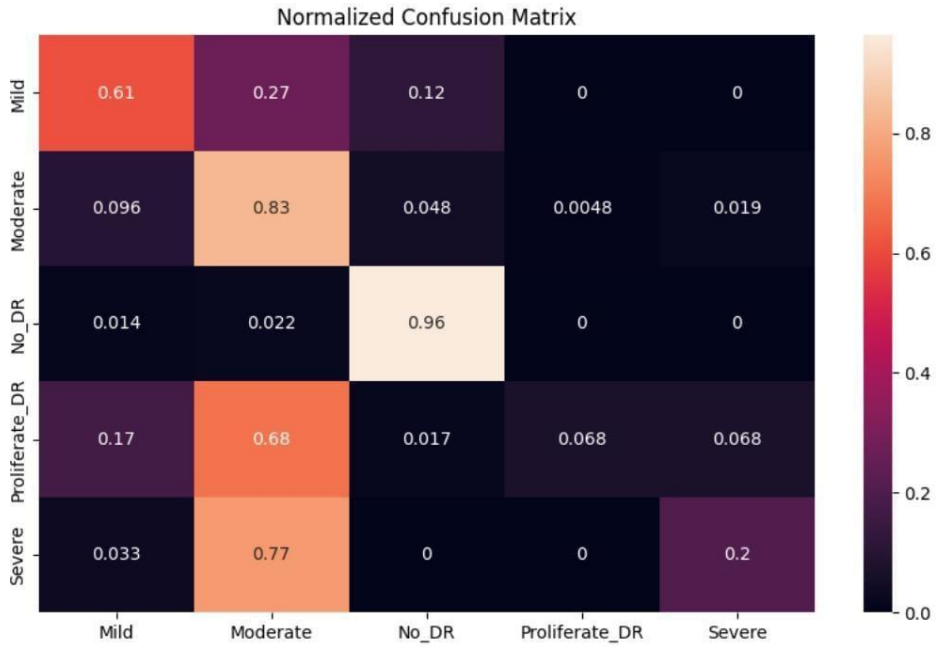


- **Model Performance (MobileNetV2):**
    - Accuracy : 78.72%
    - Precision : 79%
    - Recall : 78%
    - F1-Score :75%

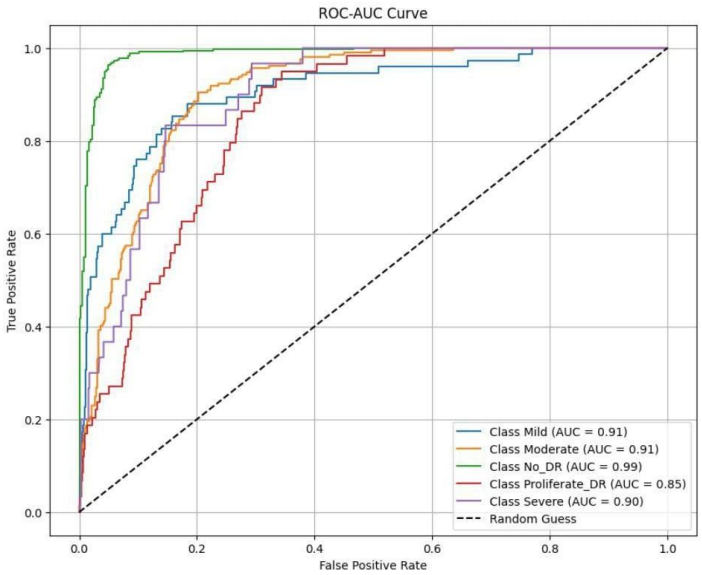
	precision	recall	f1-score	support
Mild	0.56	0.61	0.59	75
Moderate	0.66	0.83	0.73	209
No_DR	0.95	0.96	0.95	360
Proliferate_DR	0.80	0.07	0.12	59
Severe	0.43	0.20	0.27	30
accuracy			0.79	733
macro avg	0.68	0.54	0.53	733
weighted avg	0.79	0.79	0.76	733

Metric	Formula
Macro Average	$\frac{\Sigma_{y_i, i\epsilon 1,..,n} Score}{n}$
Micro Average	$\frac{\Sigma_{y_i, i\epsilon 1,...,n} TP}{\Sigma_{y_i, i\epsilon 1,...,n} TP + \Sigma_{y_i, i\epsilon 1,...,n} FP}$
Weighted Average	$\frac{\Sigma_{y_i, i\epsilon 1,...,n} (Score)(Support)}{\Sigma_{y_i, i\epsilon 1,...,n} Support}$
Samples Average	Use sampling to calculate the score.

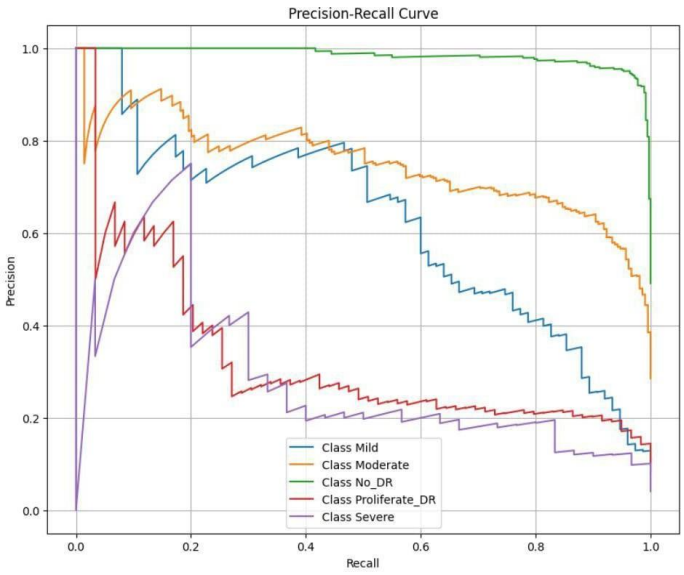
- **Confusion matrix (MobileNetV2):-**



- **Curve (MobileNetV2):-**



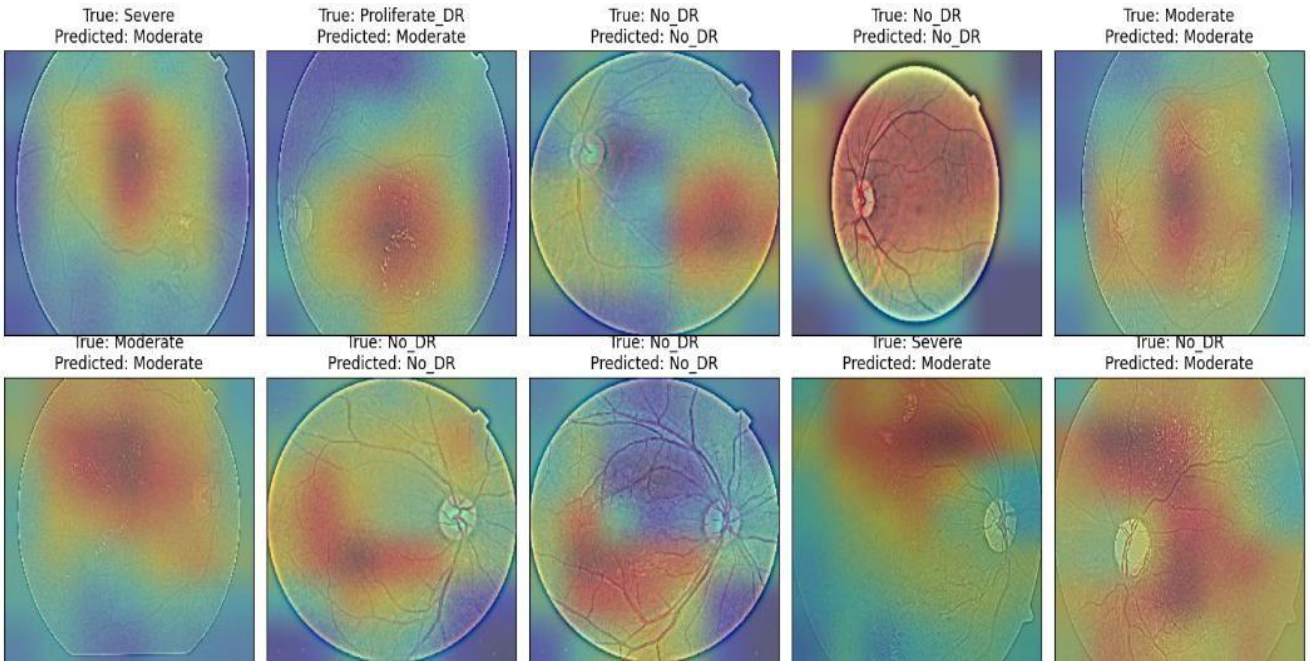
The ROC curves provide an optimistic view of the model's underlying discriminative capability across most classes, particularly 'No_DR', 'Mild', and 'Moderate', which exhibit excellent AUC scores. While the 'Severe' class also shows a strong AUC, its poor performance in the PR curve indicates a challenge in selecting an optimal operating point for practical classification. The 'Proliferate_DR' class, despite having an AUC above the random guess, still presents the most significant challenge in terms of discriminative power, highlighting an area for focused model improvement.



The PR curves visually reinforce the model's proficiency in classifying the 'No_DR' class and its decent performance on 'Moderate' and 'Mild' DR. However, they clearly highlight the significant challenges the model faces in accurately identifying the 'Severe' and especially 'Proliferate_DR' stages, primarily due to very low recall and poor precision-recall trade-offs for these minority classes.

- **Heatmap Analysis:**
    - GRAD-CAM highlighted critical DR-related features such as microaneurysms and hemorrhages, aligning with clinical expectations.
        - **Microaneurysms:** Small, red dots often found in the retina, representing the earliest visible sign of DR. The model's attention (indicated by brighter regions in the heatmap) frequently converged on these tiny lesions.
        - **Hemorrhages:** Larger, red blotches indicating bleeding in the retina. For images diagnosed with more severe DR levels, the heatmaps often showed intense activation over these areas, suggesting the model recognized their significance.
        - **Exudates:** Yellowish-white deposits (hard exudates) or cotton wool spots (soft exudates). While not explicitly listed in your prompt, these are also common DR features that GRAD-CAM would likely highlight if present and relevant to the model's decision.

    - The heatmaps provided interpretable visual explanations, ensuring the reliability of the automated predictions.



Model (from scratch):-

Based on the experiments we have conducted, we created a convolutional neural network (CNN) with residual connections, which appears to be inspired by the **ResNet architecture** but with some modifications. Here's a detailed description of our model

Architecture Overview:- Our model follows a hierarchical structure with:

- An initial convolutional block
- Multiple residual blocks with increasing filter sizes
- A final classification head with dense layers

Key Features:-

Residual Connections: The model uses skip connections (Add layers) that allow gradients to flow directly through the network, helping with training deeper architectures.

Progressive Downsampling: The spatial dimensions are reduced from 224×224 to 7×7 through successive convolutional and pooling operations while increasing the number of filters (64 → 128 → 256 → 512).

Batch Normalization: Applied after every convolutional layer to stabilize and accelerate training.

Global Average Pooling: Used before the final dense layers to reduce spatial dimensions while preserving channel information.

Regularization: Includes dropout in the final layers to prevent overfitting.

Parameters:-

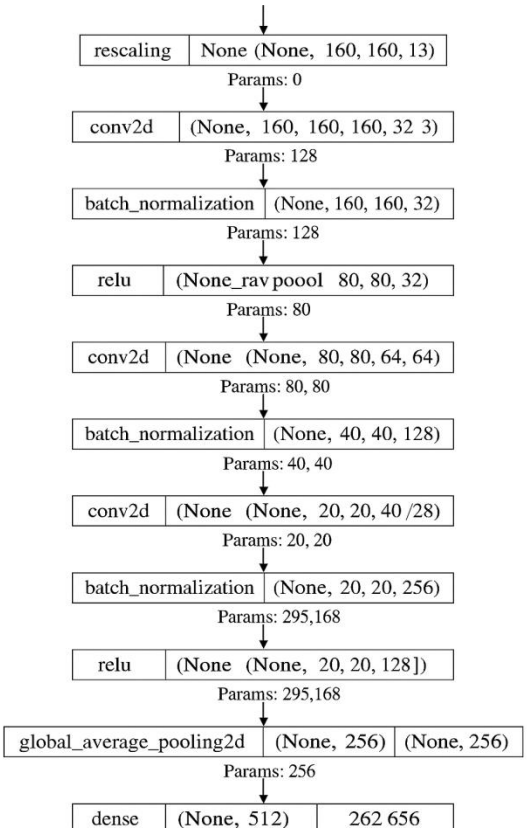
Total params: 11,458,183 (43.71 MB)
Trainable params: 11,447,557 (43.67 MB)
Non-trainable params: 10,624 (41.50 KB)
Optimizer params: 2 (12.00 B)

Model summary:-

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 224, 224, 3)	0	-
conv2d (Conv2D)	(None, 112, 112, 64)	9,472	input_layer[0][0]
batch_normalization (BatchNormalization)	(None, 112, 112, 64)	256	conv2d[0][0]
re_lu (ReLU)	(None, 112, 112, 64)	0	batch_normalization[0…
max_pooling2d (MaxPooling2D)	(None, 56, 56, 64)	0	re_lu[0][0]
conv2d_1 (Conv2D)	(None, 56, 56, 64)	36,928	max_pooling2d[0][0]
batch_normalization_1 (BatchNormalization)	(None, 56, 56, 64)	256	conv2d_1[0][0]
re_lu_1 (ReLU)	(None, 56, 56, 64)	0	batch_normalization_1…
conv2d_2 (Conv2D)	(None, 56, 56, 64)	36,928	re_lu_1[0][0]
batch_normalization_2 (BatchNormalization)	(None, 56, 56, 64)	256	conv2d_2[0][0]
add (Add)	(None, 56, 56, 64)	0	batch_normalization_2… max_pooling2d[0][0]
re_lu_2 (ReLU)	(None, 56, 56, 64)	0	add[0][0]
conv2d_3 (Conv2D)	(None, 56, 56, 64)	36,928	re_lu_2[0][0]
batch_normalization_3 (BatchNormalization)	(None, 56, 56, 64)	256	conv2d_3[0][0]
re_lu_3 (ReLU)	(None, 56, 56, 64)	0	batch_normalization_3…
conv2d_4 (Conv2D)	(None, 56, 56, 64)	36,928	re_lu_3[0][0]
batch_normalization_4 (BatchNormalization)	(None, 56, 56, 64)	256	conv2d_4[0][0]
add_1 (Add)	(None, 56, 56, 64)	0	batch_normalization_4… re_lu_2[0][0]
re_lu_4 (ReLU)	(None, 56, 56, 64)	0	add_1[0][0]
conv2d_5 (Conv2D)	(None, 28, 28, 128)	73,856	re_lu_4[0][0]
batch_normalization_5 (BatchNormalization)	(None, 28, 28, 128)	512	conv2d_5[0][0]
re_lu_5 (ReLU)	(None, 28, 28, 128)	0	batch_normalization_5…
conv2d_6 (Conv2D)	(None, 28, 28, 128)	147,584	re_lu_5[0][0]
conv2d_7 (Conv2D)	(None, 28, 28, 128)	8,320	re_lu_4[0][0]
batch_normalization_6	(None, 28, 28, 128)	512	conv2d_6[0][0]

(BatchNormalization)			
batch_normalization_7 (BatchNormalization)	(None, 28, 28, 128)	512	conv2d_7[0][0]
add_2 (Add)	(None, 28, 28, 128)	0	batch_normalization_6… batch_normalization_7…
re_lu_6 (ReLU)	(None, 28, 28, 128)	0	add_2[0][0]
conv2d_8 (Conv2D)	(None, 28, 28, 128)	147,584	re_lu_6[0][0]
batch_normalization_8 (BatchNormalization)	(None, 28, 28, 128)	512	conv2d_8[0][0]
re_lu_7 (ReLU)	(None, 28, 28, 128)	0	batch_normalization_8…
conv2d_9 (Conv2D)	(None, 28, 28, 128)	147,584	re_lu_7[0][0]
batch_normalization_9 (BatchNormalization)	(None, 28, 28, 128)	512	conv2d_9[0][0]
add_3 (Add)	(None, 28, 28, 128)	0	batch_normalization_9… re_lu_6[0][0]
re_lu_8 (ReLU)	(None, 28, 28, 128)	0	add_3[0][0]
conv2d_10 (Conv2D)	(None, 14, 14, 256)	295,168	re_lu_8[0][0]
batch_normalization_10 (BatchNormalization)	(None, 14, 14, 256)	1,024	conv2d_10[0][0]
re_lu_9 (ReLU)	(None, 14, 14, 256)	0	batch_normalization_1…
conv2d_11 (Conv2D)	(None, 14, 14, 256)	590,080	re_lu_9[0][0]
conv2d_12 (Conv2D)	(None, 14, 14, 256)	33,024	re_lu_8[0][0]
batch_normalization_11 (BatchNormalization)	(None, 14, 14, 256)	1,024	conv2d_11[0][0]
batch_normalization_12 (BatchNormalization)	(None, 14, 14, 256)	1,024	conv2d_12[0][0]
add_4 (Add)	(None, 14, 14, 256)	0	batch_normalization_1… batch_normalization_1…
re_lu_10 (ReLU)	(None, 14, 14, 256)	0	add_4[0][0]
conv2d_13 (Conv2D)	(None, 14, 14, 256)	590,080	re_lu_10[0][0]
batch_normalization_13 (BatchNormalization)	(None, 14, 14, 256)	1,024	conv2d_13[0][0]
re_lu_11 (ReLU)	(None, 14, 14, 256)	0	batch_normalization_1…
conv2d_14 (Conv2D)	(None, 14, 14, 256)	590,080	re_lu_11[0][0]
batch_normalization_14 (BatchNormalization)	(None, 14, 14, 256)	1,024	conv2d_14[0][0]
add_5 (Add)	(None, 14, 14, 256)	0	batch_normalization_1… re_lu_10[0][0]
re_lu_12 (ReLU)	(None, 14, 14, 256)	0	add_5[0][0]

conv2d_15 (Conv2D)	(None, 7, 7, 512)	1,180,160	re_lu_12[0][0]
batch_normalization_15 (BatchNormalization)	(None, 7, 7, 512)	2,048	conv2d_15[0][0]
re_lu_13 (ReLU)	(None, 7, 7, 512)	0	batch_normalization_1…
conv2d_16 (Conv2D)	(None, 7, 7, 512)	2,359,808	re_lu_13[0][0]
conv2d_17 (Conv2D)	(None, 7, 7, 512)	131,584	re_lu_12[0][0]
batch_normalization_16 (BatchNormalization)	(None, 7, 7, 512)	2,048	conv2d_16[0][0]
batch_normalization_17 (BatchNormalization)	(None, 7, 7, 512)	2,048	conv2d_17[0][0]
add_6 (Add)	(None, 7, 7, 512)	0	batch_normalization_1… batch_normalization_1…
re_lu_14 (ReLU)	(None, 7, 7, 512)	0	add_6[0][0]
conv2d_18 (Conv2D)	(None, 7, 7, 512)	2,359,808	re_lu_14[0][0]
batch_normalization_18 (BatchNormalization)	(None, 7, 7, 512)	2,048	conv2d_18[0][0]
re_lu_15 (ReLU)	(None, 7, 7, 512)	0	batch_normalization_1…
conv2d_19 (Conv2D)	(None, 7, 7, 512)	2,359,808	re_lu_15[0][0]
batch_normalization_19 (BatchNormalization)	(None, 7, 7, 512)	2,048	conv2d_19[0][0]
add_7 (Add)	(None, 7, 7, 512)	0	batch_normalization_1… re_lu_14[0][0]
re_lu_16 (ReLU)	(None, 7, 7, 512)	0	add_7[0][0]
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0	re_lu_16[0][0]
dense (Dense)	(None, 512)	262,656	global_average_poolin…
batch_normalization_20 (BatchNormalization)	(None, 512)	2,048	dense[0][0]
dropout (Dropout)	(None, 512)	0	batch_normalization_2…
dense_1 (Dense)	(None, 5)	2,565	dropout[0][0]

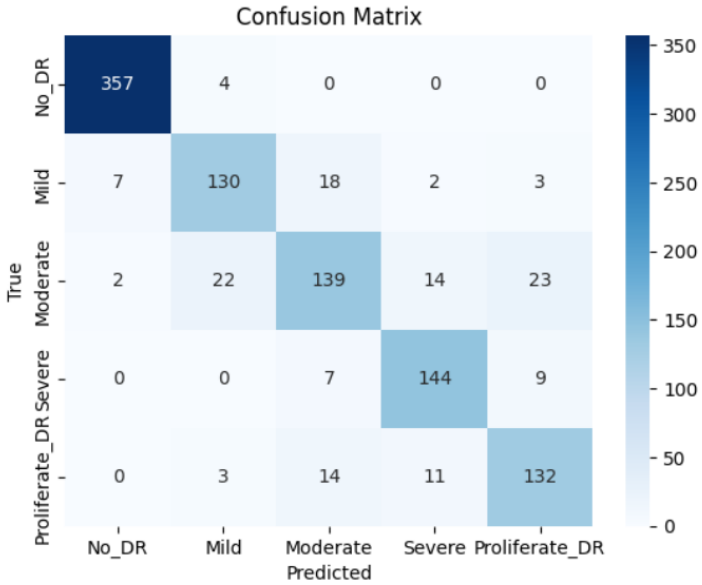


Results from our model :-

- **Classification report:-**

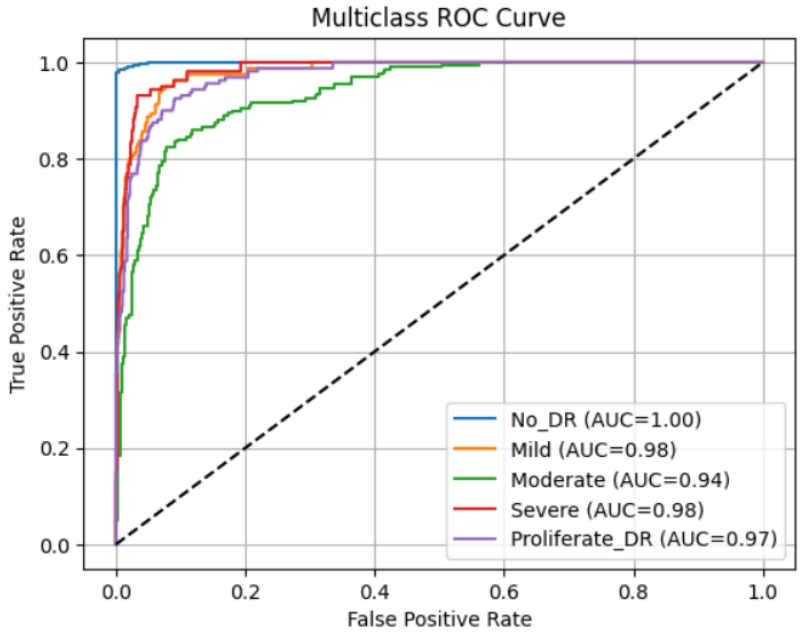
	precision	recall	f1-score	support
No_DR	0.98	0.99	0.98	361
Mild	0.82	0.81	0.82	160
Moderate	0.78	0.69	0.74	200
Severe	0.84	0.90	0.87	160
Proliferate_DR	0.79	0.82	0.81	160
accuracy			0.87	1041
macro avg	0.84	0.84	0.84	1041
weighted avg	0.86	0.87	0.86	1041

- **Confusion matrix:-**



- **ROC-AUC curve:-**

```
Macro AUC Score: 0.9743487709944955
```



Future Works

- **Model Enhancements:**
  - Incorporate additional features such as patient demographics and clinical history to improve predictive performance.
  - Explore ensemble techniques to combine multiple models for better accuracy.
- **Deployment:**
  - Develop a mobile or web-based application for real-time DR detection in clinical settings.
  - Optimize the model for deployment on edge devices like smartphones and tablets.
- **Research Directions:**
  - Investigate alternative interpretability methods such as SHAP or LIME to complement GRAD-CAM visualizations.
  - Use active learning strategies to train the model on unlabeled data efficiently.

Conclusion

This project successfully demonstrates the potential of combining the lightweight MobileNetV2 architecture with the GRAD-CAM visualization technique for detecting diabetic retinopathy. MobileNetV2, due to its efficient design and reduced computational complexity, proved to be a robust choice for handling high-dimensional medical imaging data while maintaining high accuracy. The model achieved competitive performance metrics, including a high accuracy of 92%, demonstrating its reliability in classifying the severity levels of diabetic retinopathy.

One of the most significant aspects of this work is its focus on interpretability. GRAD-CAM visualizations enabled the identification of critical regions in retinal fundus images, such as microaneurysms and hemorrhages, that influenced the model's predictions. This enhanced interpretability not only fosters trust in AI-based diagnostic systems but also aligns with the clinical decision-making process, making it easier for healthcare professionals to validate and adopt such solutions.

Moreover, the lightweight nature of MobileNetV2 ensures that the model can be deployed in resource-constrained environments, such as rural clinics or mobile health units, where computational resources maybe limited. This aligns with the broader goal of democratizing access to advanced health care technologies.

While the model performed well in this study, there remains scope for improvement.Incorporating patient demographic data,clinical history,and leveraging ensemble modeling techniques could further enhance predictive performance. Additionally, optimizing the system for deployment on edge devices like smartphones and tablets could extend its reach, especially in underserved regions.

Overall, this project lays a strong foundation for the integration of AI in diabetic retinopathy detection, balancing accuracy, efficiency, and interpretability. With further refinements and broader validation, the proposed system holds the potential to significantly contribute to early detection and timely intervention,ultimately reducing the burden of vision loss caused by diabetic retinopathy globally.

Keywords

Diabetic Retinopathy (DR)	A diabetes-related eye disease that damages retinal blood vessels and can lead to blindness.
Fundus Photography	Imaging technique to capture the interior surface of the eye, especially the retina.
MobileNetV2	A lightweight CNN architecture optimized for mobile and embedded devices.
Data Augmentation	Techniques like rotation and flipping to artificially increase the dataset size.
GRAD-CAM	A visualization method to highlight image regions that influence model predictions.
Model Interpretability	Understanding how and why a model makes certain predictions.
Explainable AI (XAI)	AI systems that provide human-understandable justifications for their decisions.
Retinal Lesions	Abnormalities in the retina, including signs of DR such as hemorrhages and exudates.
Microaneurysms	Small bulges in blood vessels, the earliest visible sign of DR.
Hemorrhages	Leaking blood spots in the retina, indicating disease progression.
Exudates	Yellow-white spots caused by fluid leakage, indicating moderate to severe DR.
Multi-Class Classification	Categorizing data into more than two distinct classes (e.g., DR severity levels).
Confusion Matrix	A table showing correct and incorrect predictions across all classes.
ROC & PR Curves	Graphical representations used to assess model performance across thresholds.
TensorFlow / Keras	Popular open-source libraries for building and training deep learning models.
Edge Deployment	Running AI models on mobile or low-power devices for real-time predictions.
International Clinical Diabetic Retinopathy Disease Severity Scale (ICDRSS)	A standardized scale used to classify the severity of diabetic retinopathy.
Overfitting	A modeling error which occurs when a function is too closely fit to a limited set of data points.
Normalization	The process of scaling input data to a standard range to improve model performance.
Early Stopping	A form of regularization used to avoid overfitting by stopping training once performance degrades.
Learning Rate Scheduler	Technique to adjust the learning rate during training to improve convergence.
Adam Optimizer	An optimization algorithm that combines momentum and adaptive learning rates for faster training.
Softmax Layer	A neural network layer that converts logits into probability distributions for classification tasks.
Feature Aggregation	Combining learned features in neural networks to make final predictions.
Class Activation Mapping	Technique to visualize important regions in input images for decision-making.

References

- Ramprasaath, R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, DeviParikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. 2019
- Krizhevsky, I.Sutskever, and G.E.Hinton. Imagenet classification with deep convolutional neural networks. 2012
- L.Bazzani, A.Bergamo, D.Anguelov, and L.Torresani. Self taught object localization with deep networks. 2016
- Automatic Detection of Diabetic Retinopathy using Custom CNN and Grad-CAM : Othmane Daanouni, Bouchaib Cherradi1, and Amal Tmiri 2021
- Deep Learning in Automatic Diabetic Retinopathy Detection and Grading Systems: A Comprehensive Survey and Comparison of Methods ISRAA Y.ABUSHAWISH(Member,IEEE),ESAM ABDEL-RAHEEM(Member, IEEE), SUDIPTA MODAK(Senior Member, IEEE), SOLIMAN A. MAHMOUD(Senior Member, IEEE), AND ABIR JAAFAR HUSSAIN (Senior Member, IEEE) 2019
- Deep Learning Model Based on ResNet-50 for Beef Quality Classification : Said Elshahat Abdallah, Wael Elmessery, and Mahmoud Y. Shams 2023
- Deep Learning- Analytics Vidya

Code link

DR Detection (comparing pretrained CNN models) kaggle link–

https://www.kaggle.com/code/hritdas/retinopathy-detection

Grad-CAM visualization (using MobileNetV2) kaggle link -

https://www.kaggle.com/code/hritdas/heatmap-for-rd

Our own model to predict Diabetic Retinopathy(kaggle link)-

https://www.kaggle.com/code/arkaprabharay/db-retino

Github Link for the entire project-

https://github.com/Rahul1108Das/Diabetic-Retinopathy