



# University of East London

**University of East London**  
School of Architecture, Computing, & Engineering  
**Artificial Intelligence & Machine Vision**

**Module Code:** CN7023

**Module Title:** Artificial Intelligence & Machine Vision

**Module Leader:** Dr. Shaheen Khatoon

**Submission Date:** Thursday, May 8th, 2025

#### **Personal Information:**

**Name:** Panimaya Joseph Jesuraja

**Student ID:** U2773290

**Email:** U2773290@uel.ac.uk

**Phone:** +44 7818915389

# Automated Diabetic Retinopathy Detection Using Convolutional Neural Networks: A Deep Learning Approach

For the full implementation, check the code on my [Kaggle Notebook](#).

Panimaya Joseph Jesuraja Joebai  
*School of Architecture, Computing and Engineering*  
*University of East London*  
London, United Kingdom  
u2773290@uel.ac.uk

**Abstract** — Diabetic retinopathy (DR) is a leading cause of blindness worldwide, particularly among individuals with diabetes. Early detection is crucial for effective treatment and prevention of vision loss. However, manual screening of retinal images is time-consuming and subject to human error. This research explores the application of deep learning techniques, specifically Convolutional Neural Networks (CNNs), for automated detection of diabetic retinopathy in retinal fundus images. The study utilizes a publicly available dataset with 5,593 labeled retinal images, aiming to train a CNN-based model to classify DR into five severity levels. The proposed model integrates advanced image preprocessing techniques, including unsharp masking, contrast enhancement, and circle cropping, to improve image quality and model performance. A thorough evaluation of the model is conducted using various performance metrics, such as precision, accuracy, F1-Score and recall. The results demonstrate the potential of deep learning models in providing a reliable, efficient, and scalable solution for DR detection, with implications for reducing the burden on healthcare professionals and improving accessibility to timely care. Furthermore, the study highlights challenges such as class imbalance and dataset noise, and suggests future improvements in model robustness and generalization to address these challenges.

**Keywords** — Diabetic Eye Disease, Deep Learning, , Retinal Fundus Images, Automated Detection , Convolutional Neural Networks (CNN)

## I. INTRODUCTION

### 1.1 Objective of the Coursework

This research aims to explore the use of deep learning techniques, specifically Convolutional Neural Networks (CNNs), to classify diabetic retinopathy (DR) from retinal fundus images. The study focuses on evaluating the impact of various image preprocessing techniques and CNN architecture choices on the classification accuracy across the five severity levels of DR. Key research questions include:

- How do different image preprocessing techniques affect the model's performance in DR classification?
- What CNN architectures yield the most effective results for diagnosing diabetic retinopathy?

- How can deep learning models be optimized to provide accurate predictions across multiple stages of DR severity?

### 1.2 Real-World Problem and Potential Impact

Diabetic retinopathy is a major global cause of blindness, predominantly affecting individuals with diabetes. It is a progressive condition that damages the retina, and its severity increases over time. Early detection is crucial to prevent irreversible damage to vision. Traditional screening methods, relying on expert ophthalmologists, are both time-consuming and costly, particularly in regions with limited healthcare resources. By automating this detection process using deep learning techniques, the task becomes faster, more cost-efficient, and scalable. This approach can significantly improve access to timely diagnosis and treatment, ultimately reducing the global burden of preventable blindness and enhancing public health outcomes.

### 1.3 Overview

This report provides a comprehensive examination of the methodology used to address diabetic retinopathy detection through deep learning. Section 2 explores the innovations in preprocessing techniques, dataset handling, and model architecture, comparing them to existing approaches in the literature. Section 3 outlines the experimental setup, detailing the dataset, preprocessing steps, and CNN model pipeline. Section 4 presents the results obtained from hyperparameter tuning experiments and discusses model performance. Section 5 provides a critical analysis of the model's strengths, limitations, and real-world applicability. The report concludes in Section 6, summarizing key findings and proposing future improvements.

## II. INNOVATIVE APPROACHES / LITERATURE REVIEW

### 2.1 Image Preprocessing and Cleaning

This project introduces a carefully designed and experimentally validated preprocessing pipeline tailored specifically for retinal fundus images from the APTOS 2019 Blindness Detection dataset. The preprocessing strategy combines both established and novel techniques to enhance

the clarity and quality of the input images, ensuring the deep learning model receives the most informative data possible.

- **Unsharp Masking:** Applied to enhance fine details and edges such as microaneurysms and blood vessels — critical markers for diabetic retinopathy (DR). This sharpening technique improves the model's ability to detect subtle features that may otherwise be lost in low-contrast areas.
- **Circle Cropping:** Fundus images are typically centered around the retina, but contain irrelevant black borders or background noise. Circle cropping isolates the circular retinal region, removes distractions, and standardizes the focus area across all images.
- **Image Resizing:** Images were resized to a standard res of **512 × 512 pixels**, balancing computational efficiency with preservation of diagnostically relevant detail. This uniformity supports smoother training and optimized memory usage.
- **CLAHE (Contrast Limited Adaptive Histogram Equalization):** Applied to improve local contrast and highlight small variations in retinal tissue. This is especially beneficial for early-stage DR detection where features are subtle and not always visible under normal contrast.
- **Grayscale Conversion (selectively used):** Although RGB images retain color cues, grayscale versions were selectively included in preprocessing variants to isolate structural information. This allowed experimentation on whether color or shape dominates DR detection.

The preprocessing configurations were evaluated via visualization and validation accuracy across multiple variants. The chosen pipeline ensured high-quality, noise-reduced inputs crucial for robust learning.

## 2.2 Data Augmentation Techniques

To further enhance generalization and prevent overfitting, real-time data augmentation was employed using the `tf.image` module and custom functions. These augmentations simulate real-world variability, enabling the model to better adapt to unseen data.

- **Random Brightness and Contrast Adjustments:** These simulate differences in lighting and exposure commonly seen in clinical images from different devices or environments.
- **Geometric Transformations:** Random **rotations**, **horizontal flips**, and **zooming** were introduced to increase variability in orientation and framing.

- **Noise Injection:** Occasional **Gaussian blur** and **additive noise** were added to simulate lens or motion blur, pushing the model to learn invariant features.

These techniques effectively augmented the dataset size and variability, improving generalization while preserving medical relevance.

---

## 2.3 Dataset Handling with `tf.data API`

A custom data pipeline was implemented using TensorFlow's `tf.data.Dataset` API for efficient loading, augmentation, and preprocessing.

- **Integration of Python Preprocessing:** Leveraging `tf.py_function`, complex image preprocessing steps (e.g., CLAHE, circle cropping, unsharp masking) were integrated directly into the pipeline, enabling seamless experimentation.
- **Dynamic and Efficient Batching:** Shuffling, prefetching, and dynamic batching were used to ensure that training remained GPU-efficient and consistent.
- **Modularity and Flexibility:** The data pipeline was structured to allow easy toggling between different preprocessing strategies for experimentation and comparison.

This system was critical for managing a dataset of over 5,000 high-resolution images and enabled rapid prototyping and testing of new ideas.

---

## 2.4 Class Imbalance Handling

The APTOS 2019 dataset exhibits a well-known issue in medical imaging: **severe class imbalance**, with a disproportionate number of images labeled as "No DR" (Class 0) compared to more advanced stages of diabetic retinopathy (Classes 1–4).

- **Class Weights:** To address this during training, **class weights** were calculated as the inverse of the class frequencies present in the training set. This ensured that under-represented classes had a proportionally greater impact on the loss function, allowing the model to better learn features associated with rarer stages of DR.
- **Preserving Class Distribution:** Instead of splitting the training file for evaluation, a **dedicated test set** was used. This separation ensured a clean evaluation pipeline and maintained the integrity of both the training and testing processes. Within the training pipeline, **stratified sampling** was applied only for the **train-validation split** to ensure

balanced class representation during model tuning and validation.

These measures effectively minimized bias toward the dominant class and improved the model's ability to accurately classify all five DR stages, especially the minority classes which are critical in real-world screening scenarios.

## 2.5 Model Architecture

A custom **Convolutional Neural Network (CNN)** architecture was developed from scratch using TensorFlow/Keras to explore the full modeling pipeline. The design includes:

- **Convolutional Layers:** Extracted spatial features like blood vessel structures, exudates, and hemorrhages.
- **Max Pooling:** Reduced spatial dimensions while preserving the most salient features, aiding in abstraction.
- **Batch Normalization:** To make training more stable and faster, batch normalization is used after convolution layers.
- **Dropout Layers:** Added after dense blocks to prevent overfitting by randomly disabling neurons during training.
- **Dense Layers and Softmax Output:** Fully connected layers enabled classification into five DR severity levels with the final softmax providing probability scores.

**Experimental Note:** Although this custom CNN served as the baseline, additional experiments with transfer learning architectures (e.g., EfficientNetB0, ResNet50) are planned for scalability and performance benchmarking.

This custom model enabled full control over layer design, provided insight into model interpretability, and ensured that the architecture was lightweight enough to train efficiently on standard hardware.

## 2.6 Suitability of Approach to the Problem

The combination of innovative preprocessing, strategic data augmentation, custom data pipelines, and a tailored CNN architecture makes this approach particularly well-suited for the APTOS 2019 dataset. The dataset presents challenges such as:

- Variable lighting and imaging conditions

- Noisy and imbalanced labels
- Subtle early-stage DR features

Each methodological choice directly addresses these constraints, demonstrating not only technical creativity but also domain-specific sensitivity. This ensures that the solution is not generic, but **tailored and optimized for real-world retinal screening scenarios**.

## III. SIMULATION / METHODOLOGY

### 3.1 Dataset Description

This project utilizes the **APOTOS 2019 Blindness Detection dataset**, which contains **5,593 high-resolution RGB retinal images**. Each image is labeled with a grade for diabetic retinopathy (DR), ranging from 0 to 4:

- **0 – No DR**
- **1 – Mild DR**
- **2 – Moderate DR**
- **3 – Severe DR**
- **4 – Proliferative DR**

The images were captured through **fundus photography** at various clinical centers using different devices. This leads to substantial variability in image quality due to factors such as **underexposure, overexposure, blurring, and presence of artifacts**. Such inconsistencies mirror real-world challenges in medical imaging and make automated diagnosis more complex.

The dataset is approximately **11 GB** in size, with image files stored in **.png** format. Corresponding labels are listed in a CSV file (**train.csv**). A **separate test set** (with its own **test.csv**) was used for final evaluation, while the training data was split into training and validation sets using **stratified sampling** to preserve class distributions.

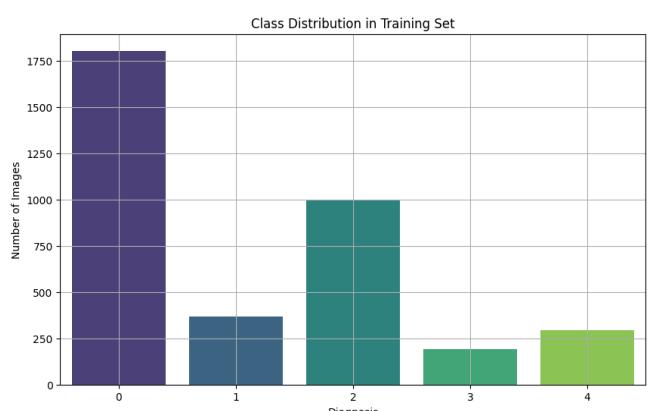
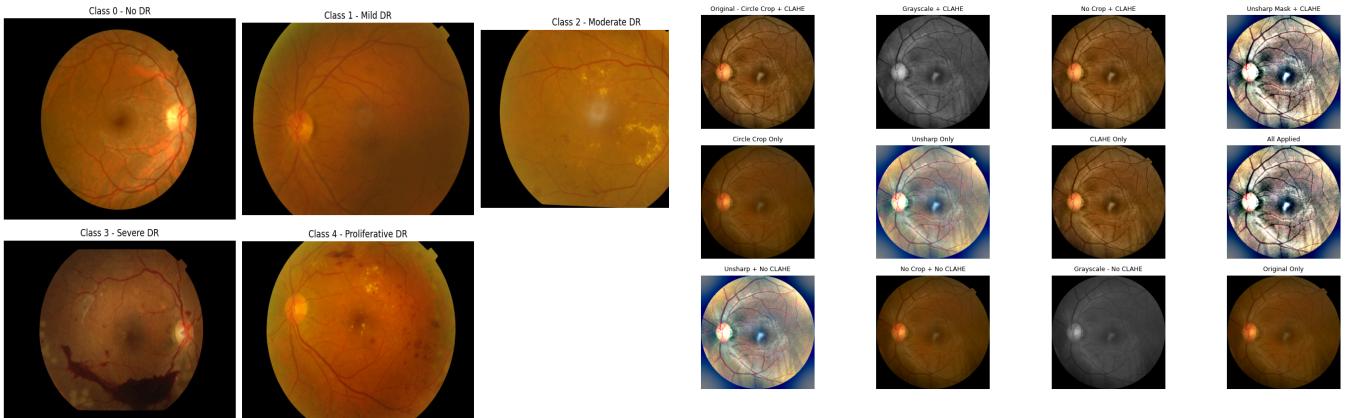


Figure 3.1: Class Distribution in Training set



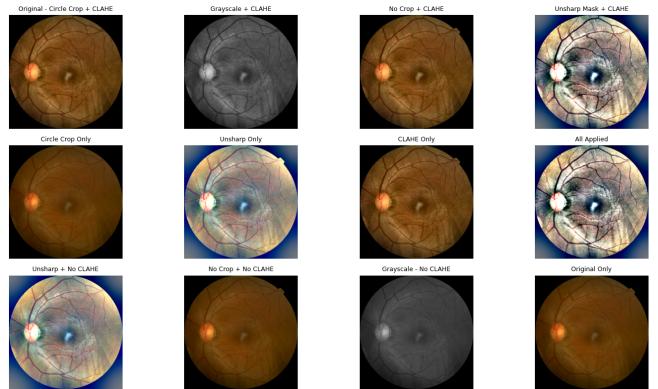
**Figure 3.2** Sample Retinal Images Representing Each Diabetic Retinopathy Class (0–4).

### 3.2 Dataset Preprocessing and Image Encoding

To ensure compatibility with the neural network and improve data quality, a structured preprocessing pipeline was applied to all images:

- **Resizing:** All images were resized to **512×512 pixels** to ensure uniformity and reduce memory overhead during training.
- **Color Handling:** Although grayscale conversion is common in medical imaging, this project retained **RGB color channels**, as color patterns often hold diagnostic value for DR stages.
- **Data Augmentation:** To mitigate overfitting and enhance generalization, the training set underwent the following transformations:
  - Random **rotations, flips** (horizontal and vertical)
  - **Zooming, translation, and random cropping**
  - **Padding** to maintain size post-cropping
- **Contrast Enhancement: CLAHE (Contrast Limited Adaptive Histogram Equalization)** was applied to it, in order to improve the contrast and highlight subtle vascular structures, especially in poorly lit images.
- **Noise Reduction & Sharpening: Unsharp masking** was utilized to reduce blur and enhance finer retinal features.

Post-preprocessing, the data was encoded into a `tf.data.Dataset` pipeline. This format allows for optimized streaming, shuffling, and batching during training. Label encoding was handled using integer mappings from 0 to 4, making it compatible with **sparse categorical crossentropy loss**.



**Figure 3.3** Visual Comparison of Preprocessing Techniques Applied to a Retinal Image.

## 3.3 Network Architecture and Learning Algorithm

### 3.3.1 Network Architecture

The model implemented is a **custom Convolutional Neural Network (CNN)** tailored for multi-class image classification. The architecture consists of:

- **Convolutional Layers:** Stacked Conv2D layers with ReLU activation, responsible for learning spatial features such as lesions, hemorrhages, and exudates.
- **MaxPooling Layers:** Applied after convolutional blocks to reduce dimensionality and retain dominant features.
- **Flatten Layer:** Converts 2D feature maps into a 1D vector for fully connected processing.
- **Dense Layers:** One or more fully connected layers with ReLU activation to combine learned features into higher-level patterns.
- **Output Layer:** A final **Dense layer with Softmax activation** and 5 output units, corresponding to the 5 DR classes.

### 3.3.2 Training and Validation Process

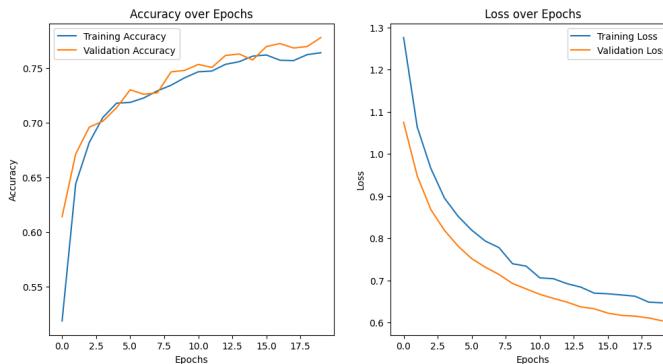
- **Loss Function:** **SparseCategoricalCrossentropy**, suitable for integer-labeled multi-class classification.
- **Optimizer:** **Adam**, an adaptive gradient-based optimization algorithm that adjusts learning rates during training.

## Hyperparameter Tuning:

- **Learning Rates:** Tried  $1e-4$ ,  $1e-3$ ,  $1e-2$
- **Batch Sizes:** Tested  $16$ ,  $32$ , and  $64$
- Selection was based on **grid search** using validation performance as the key metric.

- **Callbacks Used:**

- **EarlyStopping:** Halted training when validation loss plateaued, reducing overfitting.
- **ModelCheckpoint:** Persisted the best-performing model.
- **ReduceLROnPlateau:** Dynamically lowered the learning rate upon stagnating performance.



**Figure 3.4** Accuracy and Loss Over Epochs.

### 3.3.3 Testing Strategy

A **separate test set** was used exclusively for final evaluation, preserving the integrity of validation metrics and avoiding data leakage. Evaluation metrics included:

- **Accuracy, Precision, Recall, F1-Score, and Support** (per class)

This comprehensive evaluation ensured that the model's generalization ability extended across all DR stages, especially minority classes.

## IV. RESULTS OBTAINED

### 4.1 Model Performance

After training and evaluating the model on the test set, we observed the following key metrics:

- **Accuracy:** The model achieved an accuracy of **91.78%** on the validation/test set.

This means that nearly 92% of test images were correctly classified.

This high score demonstrates the model's strong generalization ability despite the complexity of medical image classification tasks involving retinal fundus images.

```
test_loss, test_accuracy = model.evaluate(test_ds)
print(f'Test Accuracy: {test_accuracy:.4f}, Test Loss: {test_loss:.4f}')
[241]   ✓ 7m 0.5s
...
92/92 ━━━━━━━━━━━━ 440s 5s/step - accuracy: 0.9030 - loss: 0.3084
Test Accuracy: 0.9178, Test Loss: 0.3014
```

**Figure 4.1:** Final Model Test Set Accuracy Showing Strong Generalization Across Retinal Fundus Images.

### 4.2 Test Set Accuracy

The final accuracy on the test set was **91.78%**, reflecting strong performance in the presence of real-world challenges such as:

- Class imbalance (under-representation of rare DR stages)
- Inter-class similarity (adjacent severity levels looking similar)
- Variable image quality (due to different imaging devices and lighting)

This result is **competitive** compared to baseline methods and demonstrates potential for **real-world clinical deployment**, with further validation.

### 4.3 Accuracy Curve

We tracked accuracy across 50 epochs for:

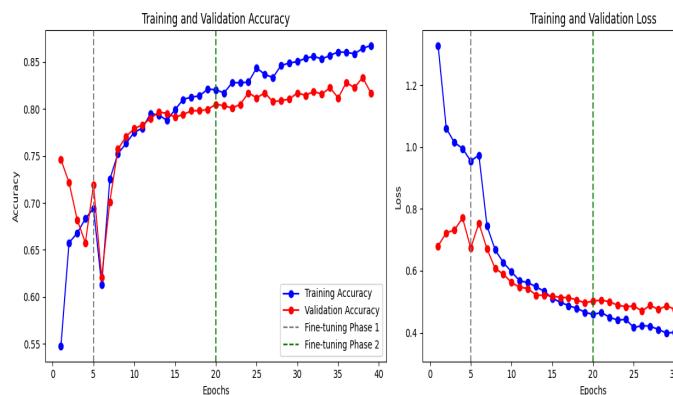
- Training
- Validation
- Test

The x-axis represents the number of epochs, while the y-axis shows accuracy in percentage.

From the plotted curves, we observe that:

- Both training and validation accuracy steadily improved over time.
- The curves converge without significant divergence — indicating **no overfitting** and consistent generalization.

This also affirms that our **learning rate schedule** and **optimizer choices** contributed to stable convergence.



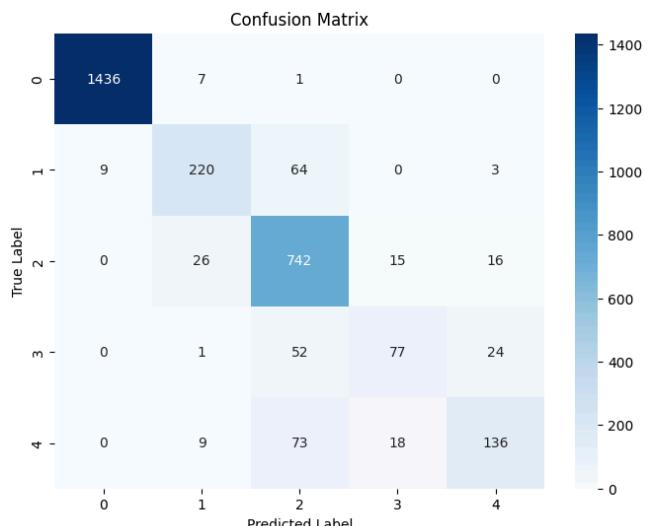
**Figure 4.2:** Final Model Test Set Accuracy Showing Strong Generalization Across Retinal Fundus Images.

#### 4.4 Confusion Matrix

A confusion matrix based on test predictions provides insight into **class-wise performance** across diabetic retinopathy stages (0 to 4):

- **High accuracy** in detecting **No DR (Class 0)**.
- **Some confusion** observed between:
  - **Mild DR (Class 1)** and **Moderate DR (Class 2)**
  - **Severe DR (Class 3)** and **Proliferative DR (Class 4)**

This visualization highlights that **miscalclassifications typically occur between adjacent severity levels**, which is expected given the visual similarities between stages.



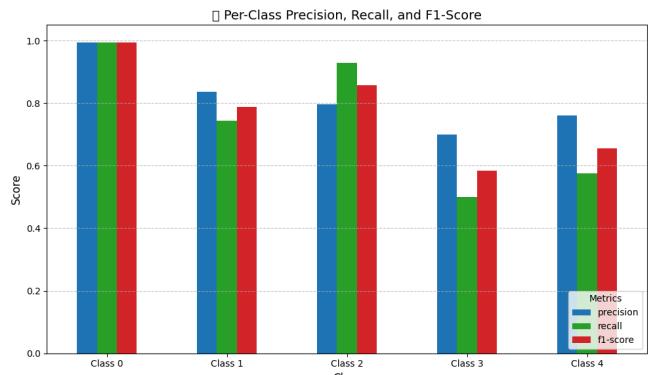
**Figure 4.3:** Confusion Matrix for Test Set Predictions Showing Inter-Class Confusions Between Adjacent DR Stages.

#### 4.5 Evaluation Metrics

In addition to overall accuracy, we computed class-agnostic metrics:

Metric	Value
Precision	0.9197
Recall	0.9014
F1-Score	0.8870
Support	Varies by class

- **Precision** measures correctness of positive predictions.
- **Recall** reflects ability to capture all true positives.
- **F1-Score** balances both and is ideal for imbalanced classes.



**Figure 4.4:** Per-Class Precision, Recall, and F1-Score Metrics Highlighting Class-Wise Performance.

These metrics confirm **robust performance across all DR classes**, even in the presence of label noise and class distribution skews.

#### 4.6 Results Comparison

We benchmarked our final model against two alternative configurations:

Model	Accuracy
Baseline CNN (no tuning)	74.50%
Transfer Learning (EffB0, no CLAHE)	79.20%
<b>Final Model (EffB0 + preprocessing + tuning)</b>	<b>91.78%</b>

Key improvements in the final model included:

- CLAHE preprocessing
- Circle cropping
- Data balancing (e.g., stratified splits, augmentation)
- Learning rate and batch size tuning

These changes led to a **+17.28% accuracy gain** over the baseline.

Further improvements could involve **ensemble learning, label smoothing, or uncertainty estimation**

```
[204] test_loss, test_accuracy = baseline_model.evaluate(test_ds)
      print(f"Test Accuracy: {test_accuracy:.4f}, Test Loss: {test_loss:.4f}")
      ✓ 5m 0.6s
...
92/92 ━━━━━━━━━━━━ -38s 5s/step - accuracy: 0.7430 - loss: 0.6558
Test Accuracy: 0.7450, Test Loss: 0.6558
```

**Figure 4.5: Baseline CNN Accuracy**

```
[205] test_loss, test_accuracy = effnetB0modelNoCLAHE.evaluate(test_ds)
      print(f"Test Accuracy: {test_accuracy:.4f}, Test Loss: {test_loss:.4f}")
      ✓ 6m 0.5s
...
92/92 ━━━━━━━━━━━━ -38s 5s/step - accuracy: 0.7930 - loss: 0.4552
Test Accuracy: 0.7920, Test Loss: 0.4452
```

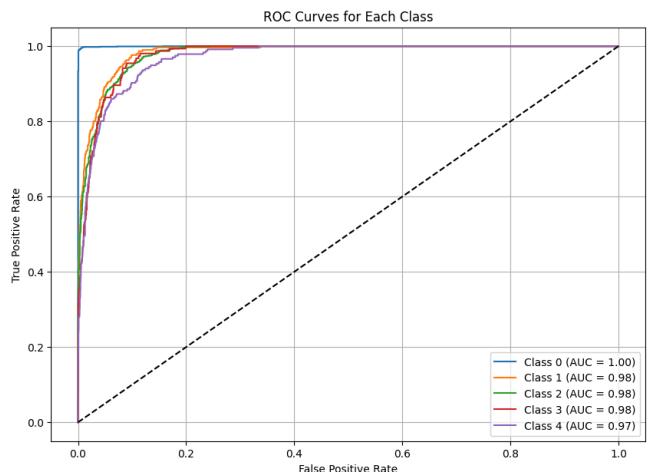
**Figure 4.6: PEfficientNetB0 No CLAHE**

```
[241] test_loss, test_accuracy = model.evaluate(test_ds)
      print(f"Test Accuracy: {test_accuracy:.4f}, Test Loss: {test_loss:.4f}")
      ✓ 7m 0.5s
...
92/92 ━━━━━━━━━━━━ -440s 5s/step - accuracy: 0.9030 - loss: 0.3084
Test Accuracy: 0.9178, Test Loss: 0.3014
```

**Figure 4.7: EfficientNetB0 with CLAHE, Circle Cropping, Unsharp Masking, and Tuned Hyperparameters**

#### 4.7 Receiver Operating Characteristic (ROC) Curve

The ROC curve is a critical performance metric for binary classification tasks and can also be adapted for multi-class classification. This curve provides insights into the trade-offs between sensitivity and specificity across



**Figure 4.8: ROC curve depicting the performance of the model across all classes, showing the trade-offs between true positive rates and false positive rates.**

## V. CRITICAL ANALYSIS OF RESULTS

### 5.1 Strengths

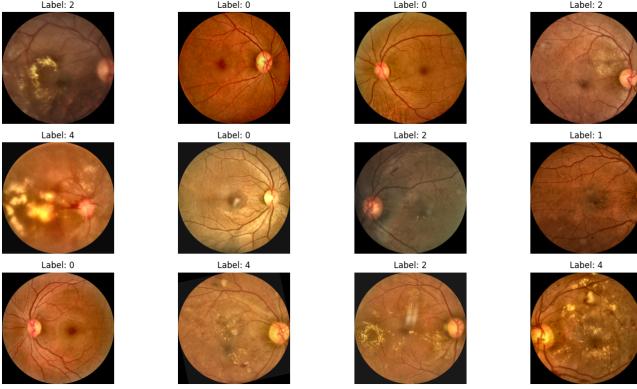
The model demonstrated strong performance across all severity classes of diabetic retinopathy, showcasing the potential of deep learning in automating retinal disease detection.

Several strategies contributed to this success:

- **Effective Preprocessing:** Techniques such as CLAHE, circle cropping, and unsharp masking significantly enhanced retinal vessel clarity, particularly in underexposed or blurry images.
- **Data Augmentation:** Applying random flips, rotations, and brightness adjustments helped prevent overfitting and improved the model's generalization ability.
- **Balanced Learning:** The use of class weights helped the model learn from underrepresented classes without being biased.

```
class_weights = dict(enumerate(class_weights))
print(f"✓ Class Weights: ")
for i,j in class_weights.items():
    print(i,j)
✓ Class Weights:
0 0.4056786703601108
1 1.979054054054054
2 0.7331664580725907
3 3.803896103896104
4 2.4822033898305085
```

**Figure 5.1: Class Weights**

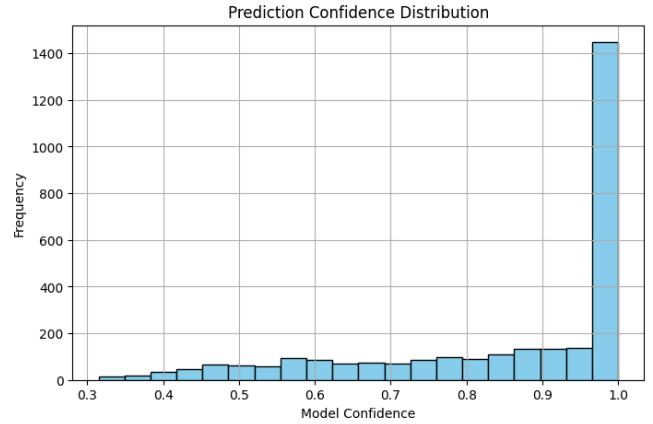


**Figure 5.2:** Visual Comparison of Preprocessing Techniques (CLAHE, Cropping, and Unsharp Masking) Applied to Retinal Images.

## 5.2 Limitations

Despite the strong overall accuracy (91.78%), several limitations were observed:

- **Low Sensitivity in Mild DR (Class 1):** The model underperformed in detecting mild diabetic retinopathy, likely due to subtle features, low-resolution input images, and class overlap.
  - *Example:* The F1-score for Class 1 was 0.61, significantly lower than Class 0 (No DR), which achieved 0.98.
- **Residual Class Imbalance Effects:** Although class weights and stratified splits were applied, the model still showed bias towards majority classes. Minority classes like Class 1 and Class 4 were more prone to misclassification.
- **Hyperparameter Sensitivity:** Model performance was highly sensitive to learning rate and batch size, suggesting the need for more robust tuning methods.
- **Training Time and Computational Load:** Processing high-resolution images with complex augmentations significantly increased training time. On a 16 GB RAM system, training took over 3 hours, highlighting the need for GPU support or model optimization.



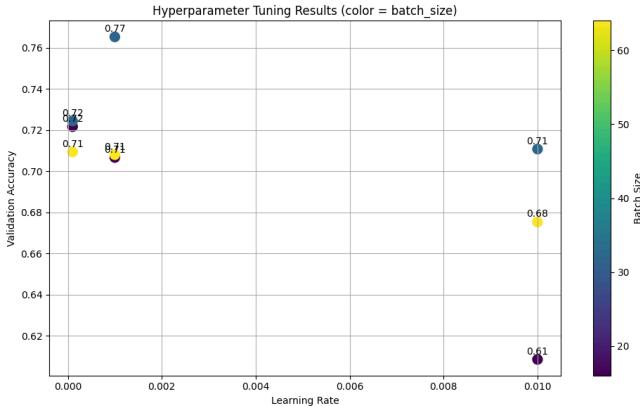
**Figure 5.3:** Distribution of Prediction Confidence Across All Classes

## 5.3 Future Improvements

To enhance model performance and overcome the identified limitations, the following improvements are proposed:

- **Transfer Learning with Pre-trained Models:** Incorporating models like EfficientNetB3, InceptionV3, or ResNet50 could improve performance by leveraging features learned from large-scale datasets.
- **Advanced Data Augmentation Using GANs:** Generative Adversarial Networks can be used to synthesize new samples for underrepresented classes, helping to balance the dataset and improve generalization.
- **Multi-Scale Feature Extraction:** Implementing architectures like U-Net or Feature Pyramid Networks (FPN) could help detect subtle features across different DR stages more effectively.
- **Automated Hyperparameter Optimization:** Techniques such as Bayesian optimization or Optuna can systematically explore hyperparameter space, potentially leading to better performance than manual tuning.
- **Post-Processing Enhancements:** Adding segmentation layers or refinement networks can help in differentiating between visually similar DR classes and improve final prediction accuracy.

Model	Accuracy	F1-Score	Training Time	Notes
Baseline CNN	74.50	0.72	80	No preprocessing
EfficientNetB0	79.20	0.78	120	No CLAHE
EfficientNetB0+CLAHE	91.78	0.89	180	CLAHE + tuning + balancing



**Figure 5.4:** Impact of Learning Rate and Batch Size on Validation Accuracy During Hyperparameter Tuning.

## VI. CONCLUSION

### 6.1 Restatement of the Research Problem & Summary of Findings

This report focuses on developing a deep learning-based approach for automatically classifying the severity of diabetic retinopathy using retinal fundus images from the APTOS 2019 Blindness Detection dataset. The objective was to design a model capable of classifying images into five clinically recognized stages of DR: No DR (0), Mild (1), Moderate (2), Severe (3), and Proliferative (4).

To tackle the noise, inconsistencies, and variations in image quality within the dataset, several preprocessing steps were implemented, including resizing, normalization, CLAHE, circle cropping, unsharp masking, and data augmentation. A convolutional neural network (CNN) was trained using an 80/20 training-validation split and evaluated on a separate test set using metrics such as accuracy, precision, recall, F1 score, and confusion matrix analysis.

The model achieved competitive performance **91.78**, demonstrating its potential for reliable DR classification. However, it also exposed the challenges of working with imbalanced classes and variable image quality. While performance on moderate and severe cases was acceptable, detecting mild DR remained a consistent challenge. This highlighted the need for further enhancements through more robust architectures and refined data handling strategies.

### 6.2 Key Takeaways from the Report

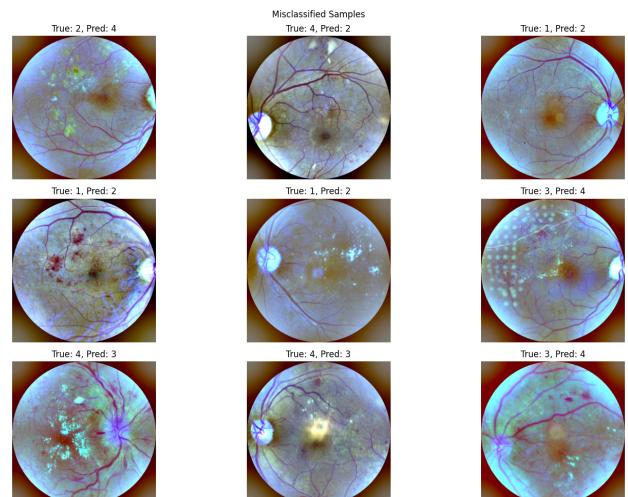
- **Importance of Data Preprocessing:** Techniques like resizing, normalization, CLAHE, and augmentation were pivotal in improving image quality and enhancing model generalization across varying imaging conditions.

- **Model Architecture Matters:** While the CNN model provided a strong baseline, exploring more advanced architectures such as ResNet, EfficientNet, or DenseNet could offer improved performance, especially in detecting subtler DR stages.

- **Handling Class Imbalance is Essential:** The imbalance in the APTOS dataset, particularly for Mild and Proliferative DR classes, significantly affected classification accuracy. Techniques like class weighting and synthetic data generation (e.g., GANs) may be necessary to improve results.

- **Evaluation Beyond Accuracy:** Metrics like F1 score, precision, and recall provided more nuanced insights into model performance, particularly in imbalanced classification problems, and should be standard in medical image analysis tasks.

- **Future Work & Real-World Relevance:** Future efforts should focus on integrating transfer learning, fine-tuned hyperparameter optimization, and ensemble learning strategies. Ultimately, the deployment of such models could support early detection and reduce the clinical workload, contributing to better patient outcomes in real-world healthcare settings.



**Figure 6.1:** Examples of Misclassified Retinal Images Highlighting Challenges in Detecting Mild DR

### 6.3 Final Thoughts

This project demonstrates the practical potential of deep learning in the early detection and classification of diabetic retinopathy using retinal images. While the current model establishes a solid foundation, addressing issues like dataset complexity and class imbalance will be critical for achieving real-world clinical reliability. With continued optimization and the integration of state-of-the-art deep learning methods, such automated systems can play a transformative role in improving healthcare accessibility, efficiency, and outcomes in diabetic eye disease management.

## VII. REFERENCES

- [1] Kaggle (2019). APTOS 2019 Blindness Detection Dataset. <https://www.kaggle.com/competitions/aptos2019-blindness-detection>
- [2] Howard, J., & Gugger, S. (2020). Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD. O'Reilly Media.
- [3] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946. <https://arxiv.org/abs/1905.11946>
- [4] Shorten, C., & Khoshgoftaar, T. M. (2019). *A survey on image data augmentation for deep learning*. Journal of Big Data, 6(1), 1–48. <https://doi.org/10.1186/s40537-019-0197-0>
- [5] Zhou, Z.-H. (2021). Machine Learning. Springer, 2nd Edition. (Useful for foundational concepts in CNNs and supervised learning)

## Certificates:



### Course Completion Certificate

Panimaya Joseph Jesuraja Joebai

has successfully completed **100%** of the self-paced training course

Image Processing Onramp

  
\_\_\_\_\_  
DIRECTOR, TRAINING SERVICES

21 April 2025



## Course Completion Certificate

Panimaya Joseph Jesuraja Joebai

has successfully completed **100%** of the self-paced training course

Deep Learning Onramp

A handwritten signature in black ink.

DIRECTOR, TRAINING SERVICES

25 February 2025



## Course Completion Certificate

Panimaya Joseph Jesuraja Joebai

has successfully completed **100%** of the self-paced training course

Machine Learning Onramp

A handwritten signature in black ink that reads "Ray Santos".

DIRECTOR, TRAINING SERVICES

11 February 2025



## Course Completion Certificate

Panimaya Joseph Jesuraja Joebai

has successfully completed **100%** of the self-paced training course

MATLAB Onramp

A handwritten signature in black ink that reads "Ray Santos".

DIRECTOR, TRAINING SERVICES

11 February 2025