Diabetic Retinopathy Classification Using Deep Learning

DL CCP

Dawood Shahzad (034)

Hassam Azam Siddiqui (035)

Bilal Ahmed (028)

Contents

Introduction

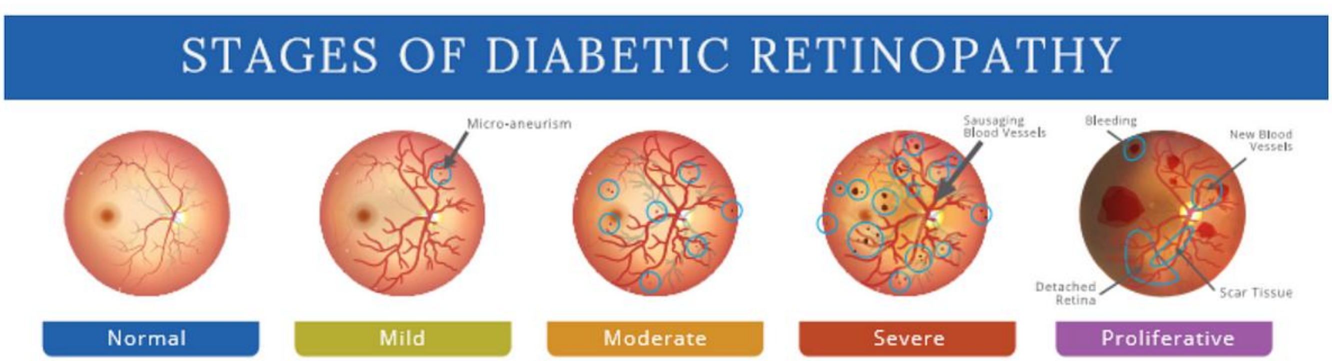
Diabetic retinopathy (DR) is a diabetes complication that affects the eyes, potentially leading to blindness if left untreated. Early detection and classification of DR stages are crucial for effective treatment. This report presents a deep learning approach to classify retinal images into different stages of DR using state-of-the-art architectures: ConvNeXt.

The study aims to analyze and compare the performance of these models in terms of classification accuracy, robustness, and generalization to demonstrate the impact of recent innovations in deep learning. By implementing and comparing these state-of-the-art techniques, we provide insights into how they enhance traditional architectures, ensuring better outcomes in the early and accurate detection of Diabetic Retinopathy.

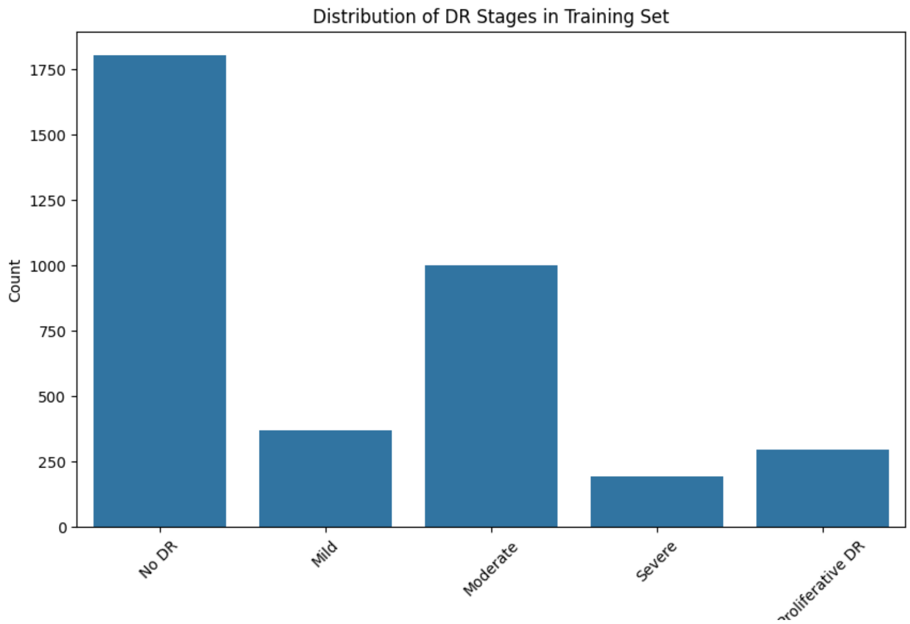
Dataset

The APTOS 2019 Blindness Detection dataset was used for this project, which contains:

- 3,662 training images
- 1,928 test images
- 5 classes of DR stages:
    a. No DR
    b. Mild
    c. Moderate
    d. Severe
    e. Proliferative DR

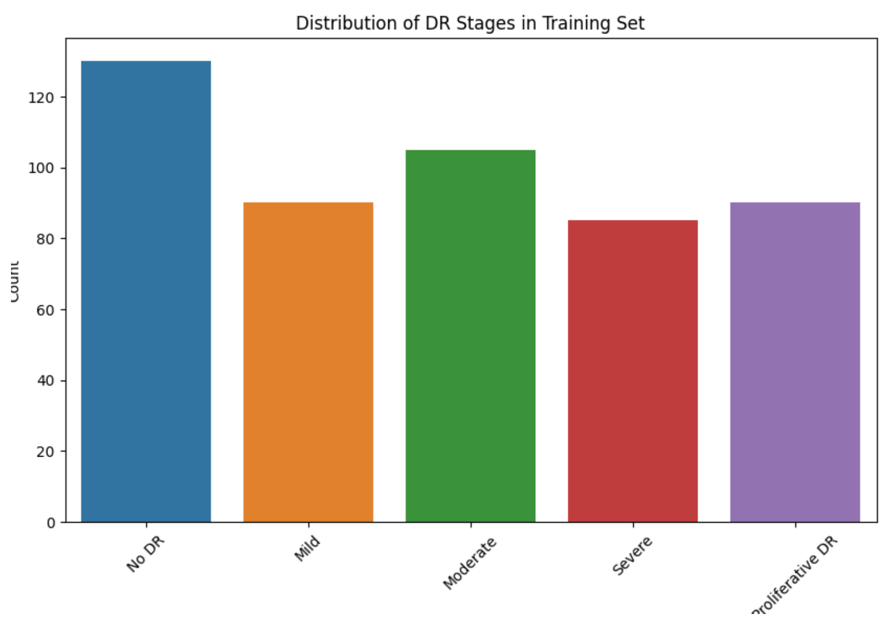


The dataset had class imbalance



This class imbalance was handled by selecting a curated sample of images from each class, keeping it close to equal distribution

Final distribution was like this



Original Training Dataset: 3662 Images

Balanced Training Dataset: 1500 Images

Testing Dataset: 600 Images (from the leftover training images)

Model 01: - ConvNeXt –

Approach

-Data Preprocessing

The following preprocessing steps were applied:

- Image resizing to 224x224 pixels
- Normalization using ImageNet statistics
- Data augmentation techniques:
  - Horizontal and vertical flips
  - Random rotations
  - Elastic transformations
  - Brightness and contrast adjustments

Model Architecture

-ConvNeXt

ConvNeXt-tiny was used as the base model with the following modifications:

- Pretrained weights enabled
- Custom classification head:
  - Linear layer (1280 → 512)
  - ReLU activation
  - Dropout (0.3)
  - Linear layer (512 → 5)

Training Process

-Training Configuration

The model was trained with the following settings:

- Optimizer: AdamW
- Learning rate: 1e-4
- Weight decay: 1e-4
- Batch size: 32
- Number of epochs: 10
- Learning rate scheduler: ReduceLROnPlateau
- Loss function: Cross Entropy Loss

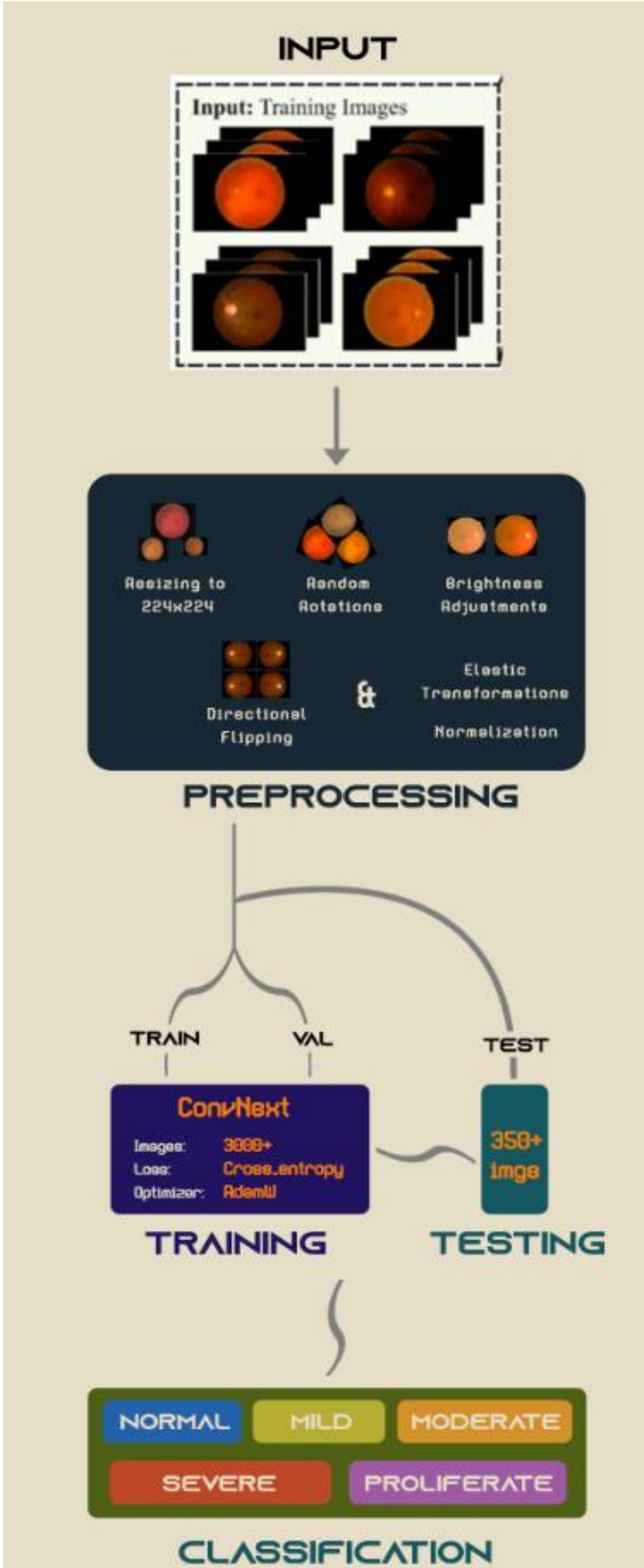
-Training Results

The models were evaluated based on:

- Training accuracy
- Validation accuracy
- Confusion matrix
- Class-wise performance metrics

Model checkpoints were saved based on validation accuracy to prevent overfitting.

Workflow


INPUT
Input: Training Images
Resizing to 224x224
Random Rotations
Brightness Adjustments
Directional Flipping
&
Elastic Transformations
Normalization
PREPROCESSING
TRAIN
VAL
TEST
ConvNext
Images: 3000+
Loss: Cross_entropy
Optimizer: AdamW
350+ imgs
TRAINING
TESTING
NORMAL
MILD
MODERATE
SEVERE
PROLIFERATE
CLASSIFICATION

Model 02: - ConvNeXt + CNN –

Approach

-Data Preprocessing

The same preprocessing steps were applied as the previous model:

- Image resizing to 224x224 pixels
- Normalization using ImageNet statistics
- Data augmentation techniques:
    - Horizontal and vertical flips
    - Random rotations
    - Elastic transformations
    - Brightness and contrast adjustments

Hybrid Model Architecture

-ConvNeXt x CNN

ConvNeXt-tiny was used as the base model with the following modifications:

- Pretrained weights enabled (IMAGENET1K_V1)
- ConvNext used as a feature extractor
- Custom classification head:
    - Linear layer (768 → 512)
    - Conv2d Blocks:
        - Batch Normalization
        - ReLU activation
        - Dropout (0.3)
    - Adaptive Average Pooling
    - Linear layer (256 → 5)

Training Process

-Training Configuration

The model was trained with the same settings as the previous model:

- Optimizer: AdamW
- Learning rate: 1e-4
- Weight decay: 1e-4
- Batch size: 32
- Number of epochs: 10
- Learning rate scheduler: ReduceLROnPlateau
- Loss function: Cross Entropy Loss

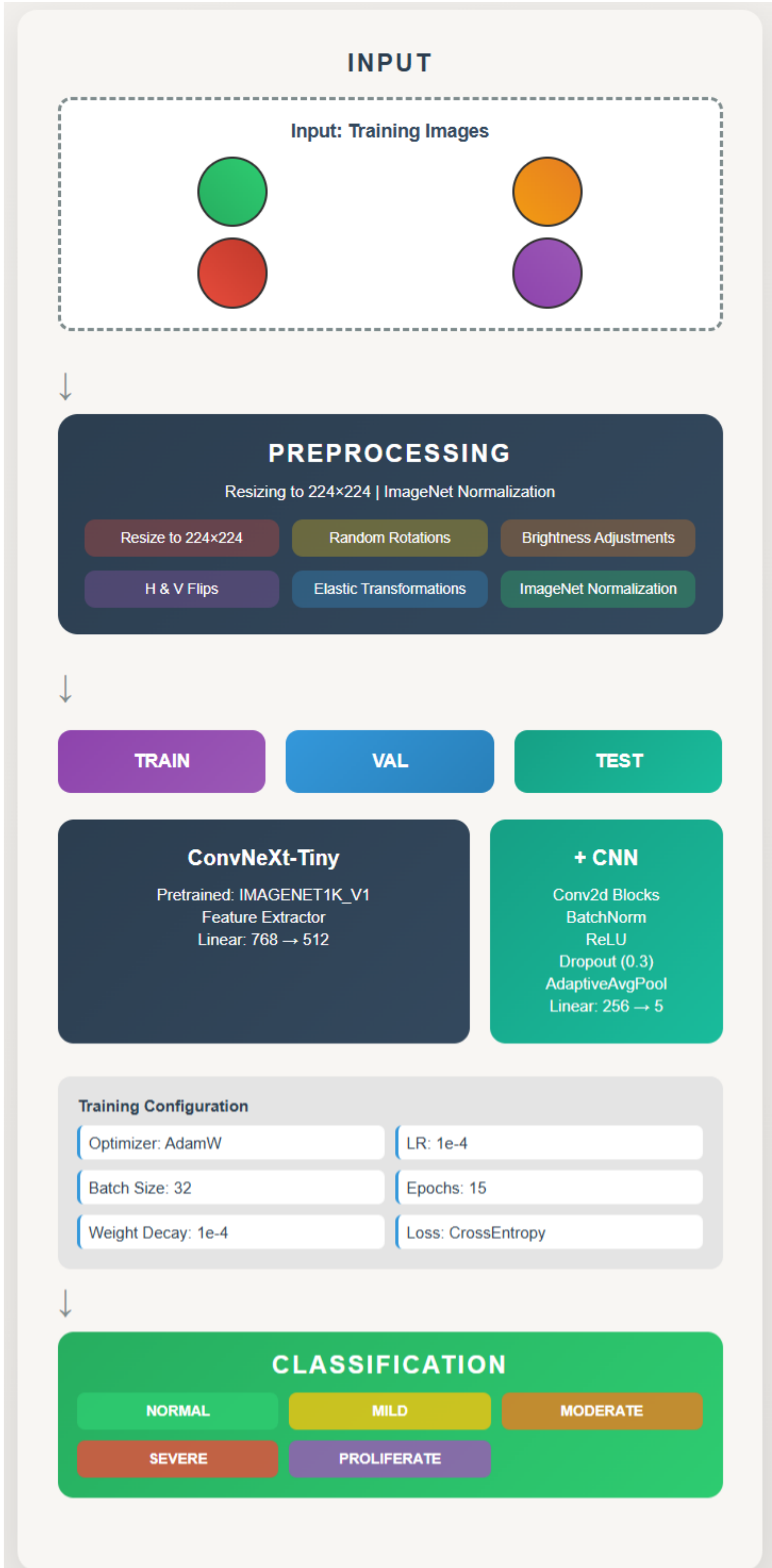
-Training Results

The models were evaluated based on:

- Training accuracy
- Validation accuracy
- Confusion matrix
- Class-wise performance metrics

Model checkpoints were saved based on validation accuracy to prevent overfitting.

Workflow


INPUT
Input: Training Images
PREPROCESSING
Resizing to 224×224 | ImageNet Normalization
Resize to 224×224
Random Rotations
Brightness Adjustments
H & V Flips
Elastic Transformations
ImageNet Normalization
TRAIN
VAL
TEST
ConvNeXt-Tiny
Pretrained: IMAGENET1K_V1
Feature Extractor
Linear: 768 → 512
+ CNN
Conv2d Blocks
BatchNorm
ReLU
Dropout (0.3)
AdaptiveAvgPool
Linear: 256 → 5
Training Configuration
Optimizer: AdamW
LR: 1e-4
Batch Size: 32
Epochs: 15
Weight Decay: 1e-4
Loss: CrossEntropy
CLASSIFICATION
NORMAL
MILD
MODERATE
SEVERE
PROLIFERATE

Comparative Analysis

Calibration				
Model	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
ConvNext	0.4798	82.20%	0.4000	84.64%
ConvNext + CNN	0.0079	91.71%	0.0192	81.42%

Testing				
Model	Precision	Recall	F1-Score	Test Accuracy
ConvNext	0.86	0.85	0.84	84.97%
ConvNext + CNN	0.85	0.84	0.84	84.15%

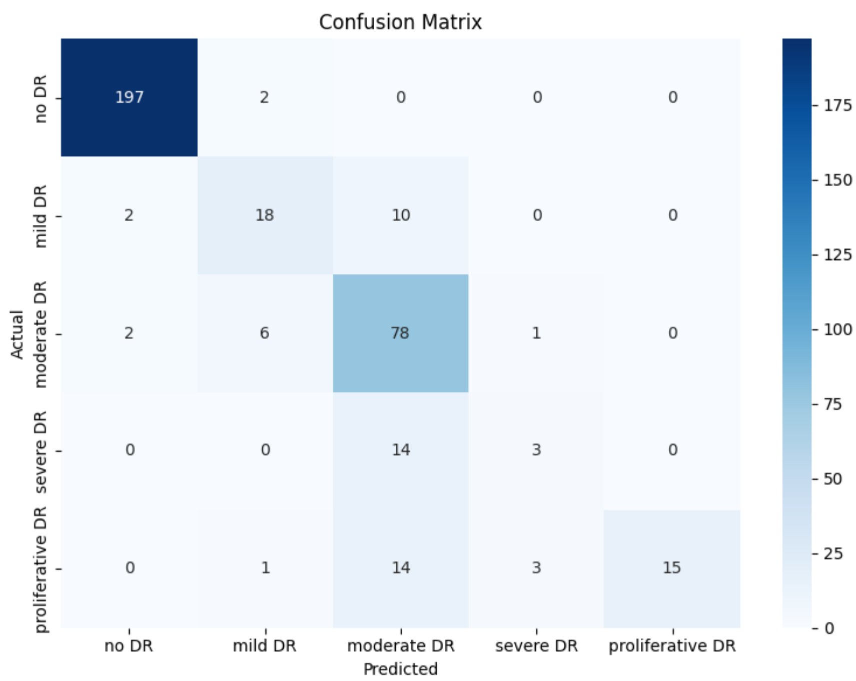
```
Test Accuracy: 84.97%

Classification Report:
                  precision    recall  f1-score   support

           no DR       0.98      0.99      0.98       199
         mild DR       0.67      0.60      0.63        30
     moderate DR       0.67      0.90      0.77        87
       severe DR       0.43      0.18      0.25        17
proliferative DR       1.00      0.45      0.62        33

        accuracy                           0.85       366
       macro avg       0.75      0.62      0.65       366
    weighted avg       0.86      0.85      0.84       366
```

Model 01

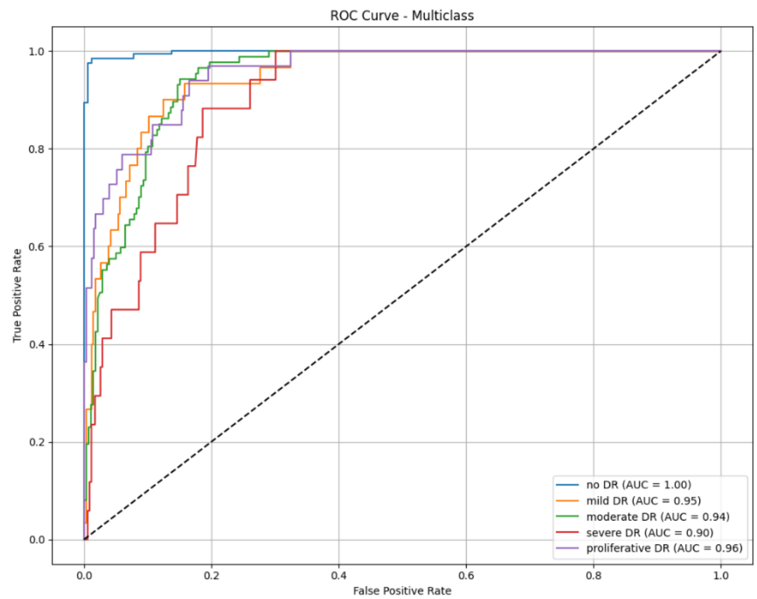
```
Test Accuracy: 84.15%

Classification Report:
                  precision    recall  f1-score   support

           no DR       0.98      0.98      0.98       199
         mild DR       0.64      0.53      0.58        30
     moderate DR       0.71      0.83      0.76        87
       severe DR       0.37      0.41      0.39        17
proliferative DR       0.89      0.52      0.65        33

        accuracy                           0.84       366
       macro avg       0.72      0.65      0.67       366
    weighted avg       0.85      0.84      0.84       366
```

Model 02

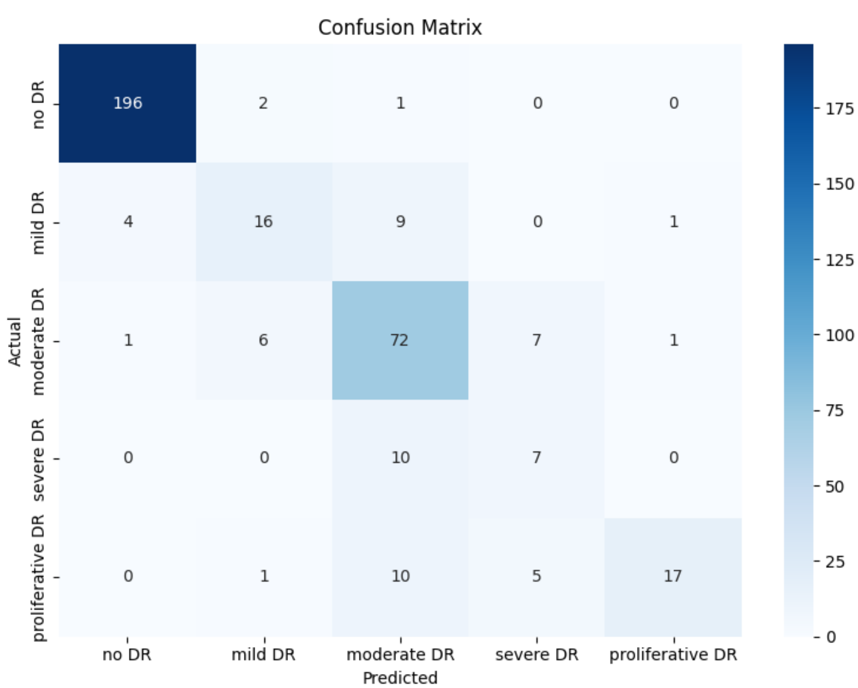
Confusion matrix (Model 01):



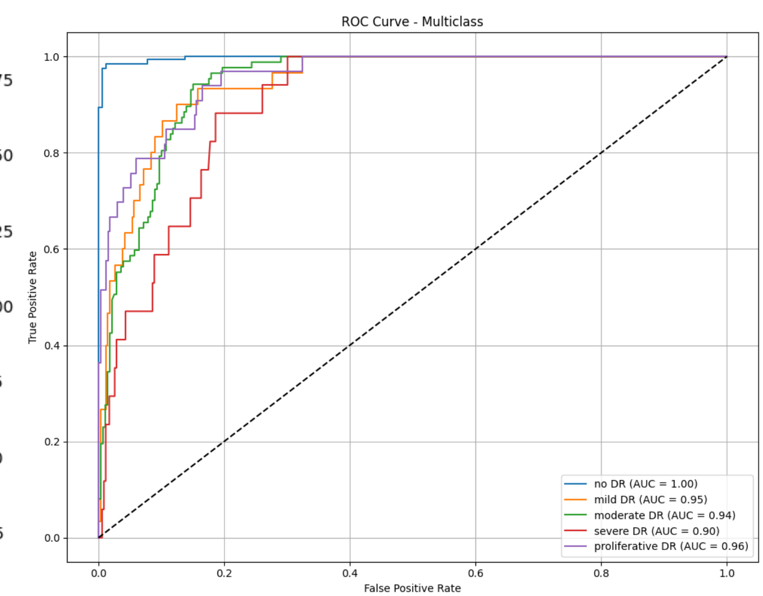
ROC Curve (model 01)



Confusion matrix (Model 02 Hybrid):



ROC Curve (model 02)



Enhancements Over Traditional Architectures

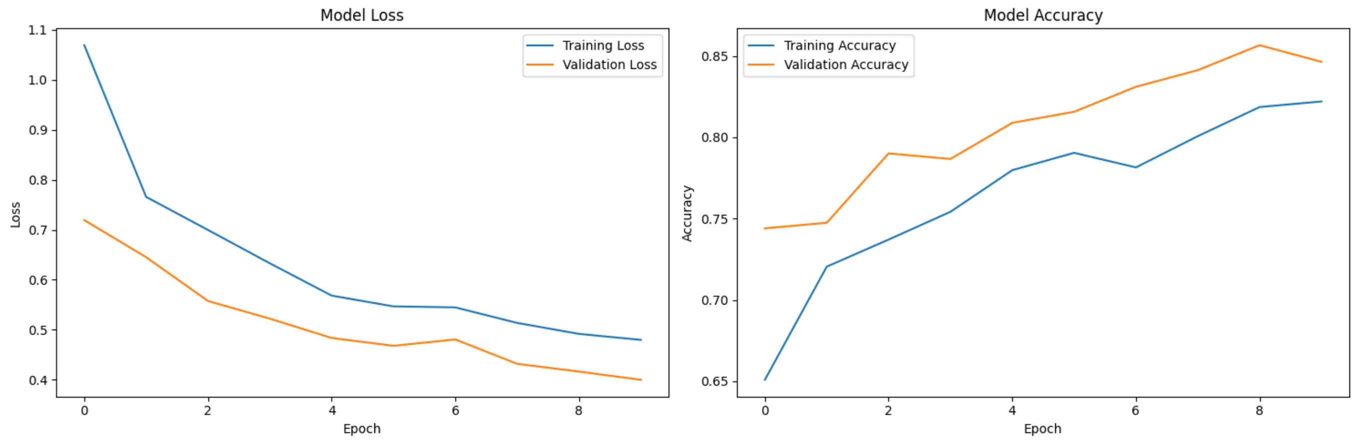
ConvNeXt Innovations

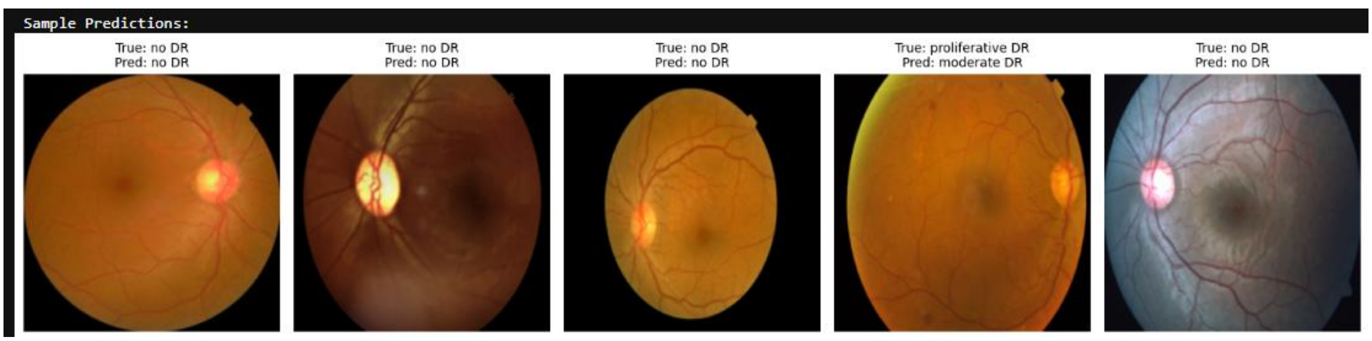
ConvNeXt brings several key improvements over traditional CNN architecture:

- Modernized ResNet Design:
  - ➢ ConvNeXt re-examines and updates the design choices in the classic ResNet architecture, incorporating modern techniques to enhance performance.
- Depthwise Convolutions:
  - ➢ It uses depthwise convolutions, which are more efficient than standard convolutions. This helps in reducing the number of parameters and computational complexity while maintaining performance.
- LayerNorm Instead of BatchNorm:
  - ➢ It replaces Batch Normalization (BatchNorm) with Layer Normalization (LayerNorm). LayerNorm is more stable and performs better in various settings, especially with smaller batch sizes.
- GELU Activation Function:
  - ➢ It uses the Gaussian Error Linear Unit (GELU) activation function instead of the traditional ReLU. GELU provides smoother gradients and often leads to better performance in deep networks.
- Stochastic Depth:
  - ➢ It incorporates stochastic depth, a regularization technique that randomly drops layers during training. This helps in preventing overfitting and improves generalization.

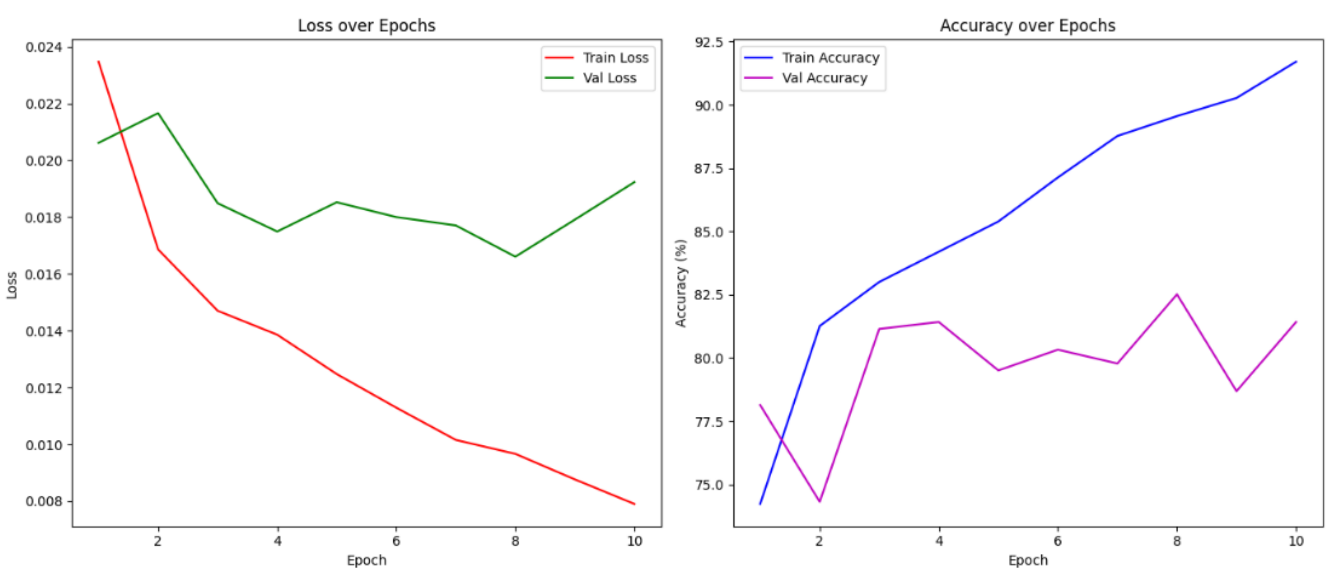
Results

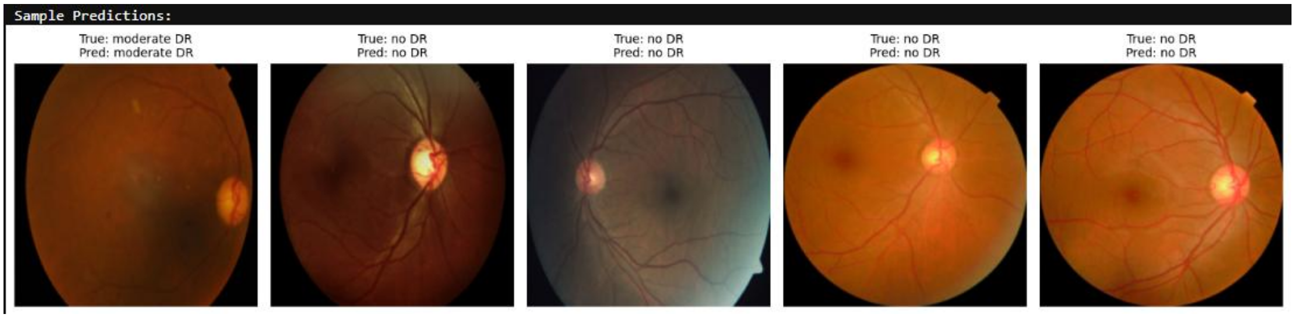
ConvNeXt Training Performance:





ConvNeXt + CNN Training (Hybrid) Performance:





Conclusion

This study demonstrates the significant impact of modern deep learning architectures in diabetic retinopathy classification. Through our comprehensive evaluation, ConvNeXt has proven to be a good choice for this critical medical imaging task. The model's exceptional performance, combined with its practical advantages, makes it an ideal solution for real-world clinical applications.

References

1. Z. Liu, H. Mao, C. -Y. Wu, C. Feichtenhofer, T. Darrell and S. Xie. A ConvNet for the 2020s. (2022).
2. K. O'Shea, R. Nash. An Introduction to Convolutional Neural Networks (2015)
3. APTOS 2019 Blindness Detection Dataset.
4. Diabetic Retinopathy Detection Guidelines.

Code Implementation

The complete implementation is available in the Jupyter notebook, including:

- Data preprocessing pipeline
- Model architecture
- Training loops
- Evaluation metrics
- Prediction functions