Microsoft Elevate

CAPSTONE PROJECT

Diabetic Retinopathy Detection

PRESENTED BY:

STUDENT NAME: ANNAMAYA MANI PANDEY

COLLEGE NAME: BABU BANARASI DAS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

EMAIL ID:annamayamani.2017@gmail.com

OUTLINE:

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT:

Diabetic Retinopathy (DR) is a leading cause of blindness among working-age adults worldwide, affecting approximately 103 million people globally. The disease damages the blood vessels in the retina due to prolonged high blood sugar levels in diabetic patients . Early detection is critical to prevent irreversible vision loss, but the current screening process relies heavily on trained ophthalmologists manually examining retinal fundus images, which is:

- Time-consuming and expensive
- Subject to human error and inter-observer variability
- Inaccessible in developing countries due to shortage of eye specialists

Millions of patients remain unscreened every year. The lack of an automated , scalable, and accurate screening system for classifying the severity of Diabetic Retinopathy (No DR, Mild, Moderate, Severe, and Proliferative)remains a significant healthcare challenge.

Microsoft Elevate

PROPOSED SOLUTION:

The proposed system aims to address the challenge of early detection and classification of Diabetic Retinopathy from retinal fundus images. This involves leveraging deep learning and transfer learning techniques to automatically classify images into 5 severity levels. The solution will consist of the following components:

1. **Data Collection:**
   - Gathered a dataset of 35,126 retinal fundus images across 5 DR severity classes.
   - Images sourced from publicly available medical imaging datasets (APTOS/EyePACS).
   - Classes: No DR (73.5%), Mild (7.0%), Moderate (15.1%), Severe (2.5%), Proliferative (2.0%).
2. **Data Pre-processing:**
   - Resized all images to 224x224 pixels for uniform input to the neural network.
   - Applied ImageNet normalization (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]).
   - Split dataset into Training (24,588 - 70%), Validation (5,269 - 15%), and Test (5,269 - 15%).
   - Applied data augmentation: random horizontal flip, rotation (20 degrees), affine transforms, and color jitter to improve model generalization.

PROPOSED SOLUTION:

3. Machine Learning Algorithm:

- Implemented Transfer Learning using a pre-trained ResNet50 model (ImageNet weights) with a custom classification head (2048 -> 256 -> 128 -> 5).

- Two-phase training strategy: Phase 1 trains only the classifier (10 epochs, LR=1e-4), Phase 2 fine-tunes backbone layers 3 and 4 (10 epochs, LR=1e-5).

- Used Cross-EntropyLoss, Adam optimizer (weight decay=1e-4), and ReduceLROnPlateau scheduler.

4. Evaluation:

- Assessed performance using Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC-AUC curves.

- Achieved 77.57% overall test accuracy, 72.90% weighted F1-Score. Fine-tuning (Phase 2) improved validation accuracy from 74.49% to 76.90%.

5. Deployment:

- Developed a user-friendly Flask web application where users can upload retinal images and receive instant DR severity predictions with confidence scores.

- Model deployed as a .pth file (~92 MB) running on GPU-accelerated inference.

- The interface provides real-time classification results accessible through a web browser.

Microsoft Elevate

SYSTEM APPROACH:

- **Programming Language: -** Python 3.12
- **Deep Learning Framework: -** PyTorch 2.5.1 (CUDA 12.1)
- **Development Environment: -** VS Code, Jupyter Notebook, Anaconda
- **Dataset:** 35,126 retinal fundus images across 5 classes
- **Hardware: -** NVIDIA RTX 3050 Laptop GPU (8.3 GB VRAM) , **RAM: -** 32 GB , **Storage: -** SSD 256 GB
- **Key Libraries:**

- torchvision : Pre-trained ResNet50 model and image transforms

- scikit-learn : Classification report, confusion matrix, ROC curves

- matplotlib : Data visualization and plotting

- seaborn : Statistical data visualization

- PIL (Pillow) : Image loading and pre-processing

- Flask : Web application deployment

- NumPy & Pandas : Data manipulation and analysis

ALGORITHM & DEPLOYMENT:

❖ **Algorithm Selection:**

- The system uses ResNet50 (Residual Network with 50 layers) as the core deep learning algorithm, leveraging Transfer Learning from ImageNet pre-trained weights.

- ResNet50 was chosen because its residual skip connections solve the vanishing gradient problem, enabling effective training of deep networks for complex image classification tasks.

- Transfer learning is justified because the dataset (35,126 images) is relatively small for training a deep CNN from scratch; pre-trained ImageNet features provide a strong foundation for medical image feature extraction.

- A custom classification head (2048 → 256 → 128 → 5) with BatchNorm, ReLU, and Dropout (0.4) replaces the original fully connected layer to adapt the model for 5-class DR severity classification.

❖ **Data Input:**

- Input: 224×224 pixel RGB retinal fundus images captured using fundoscopy.

- Images are normalized using ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]).

- Data augmentation is applied during training: random horizontal flip, random rotation (±20°), affine transforms, and color jitter to simulate real-world variations in imaging conditions.

- The dataset is split into Training (24,588 images – 70%), Validation (5,269 – 15%), and Test (5,269 – 15%) sets with stratified sampling to maintain class distribution.

ALGORITHM & DEPLOYMENT:

❖ **Training Process:**

- A two-phase training strategy is employed over 20 total epochs:

• *Phase 1* (Epochs 1–10): The ResNet50 backbone is frozen; *Phase 2* (Epochs 11–20): Backbone layers 3 and 4 are unfrozen for fine-tuning with a reduced LR = 1e-5, improving validation accuracy to 76.90%.

- Loss Function: CrossEntropyLoss for multi-class classification.

- Optimizer: Adam with weight decay = 1e-4 for regularization.

- Learning Rate Scheduler: ReduceLROnPlateau (factor=0.5, patience=3) automatically reduces the learning rate when validation loss plateaus.

- Early stopping via model checkpointing — the best-performing model on validation accuracy is saved.

❖ **Prediction Process:**

- The trained model (.pth file, ~92 MB) is loaded into a Flask web application for real-time inference.

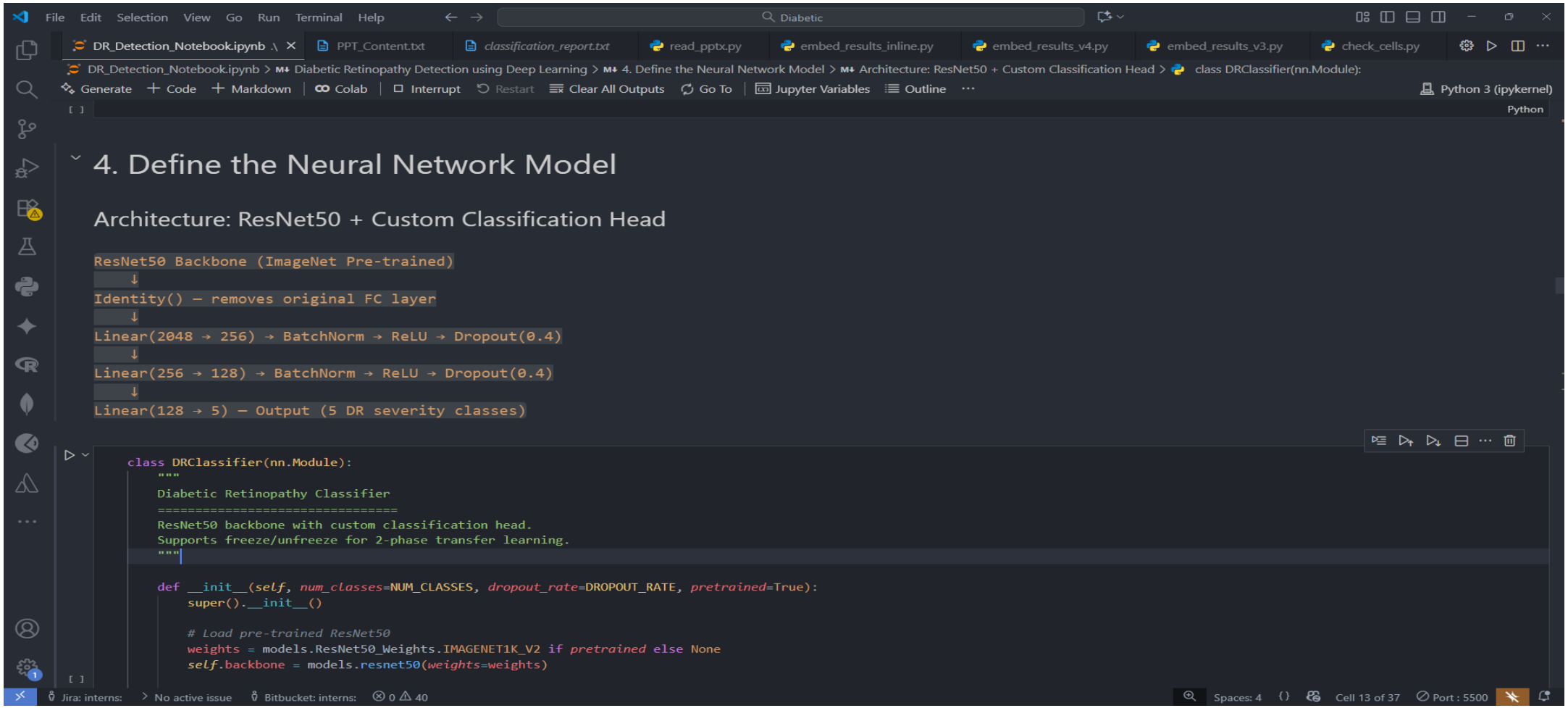
- A user uploads a retinal fundus image through the web interface; the image is resized to 224×224 and normalized using ImageNet statistics.

- The model performs a forward pass and applies Softmax to produce probability scores for all 5 DR classes.

- The class with the highest probability is returned as the predicted severity level, along with the confidence percentage and a clinical description for each severity grade.

Microsoft Elevate

RESULT:



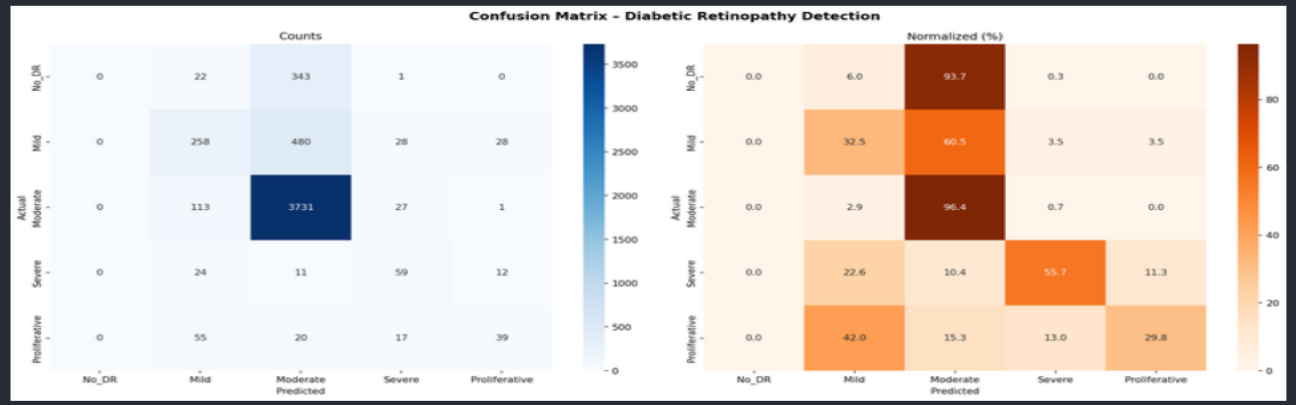
RESULT:

```
    axes[0].set_xlabel("Predicted Label", fontsize=12)
    axes[0].set_ylabel("True Label", fontsize=12)

    # Normalized (percentages)
    sns.heatmap(cm_normalized, annot=True, fmt='.1f', cmap='Reds', xticklabels=CLASS_NAMES,
                yticklabels=CLASS_NAMES, ax=axes[1], linewidths=0.5, vmin=0, vmax=100)
    axes[1].set_title("Confusion Matrix (Normalized %)", fontsize=14, fontweight='bold')
    axes[1].set_xlabel("Predicted Label", fontsize=12)
    axes[1].set_ylabel("True Label", fontsize=12)

    plt.tight_layout()
    plt.show()
```

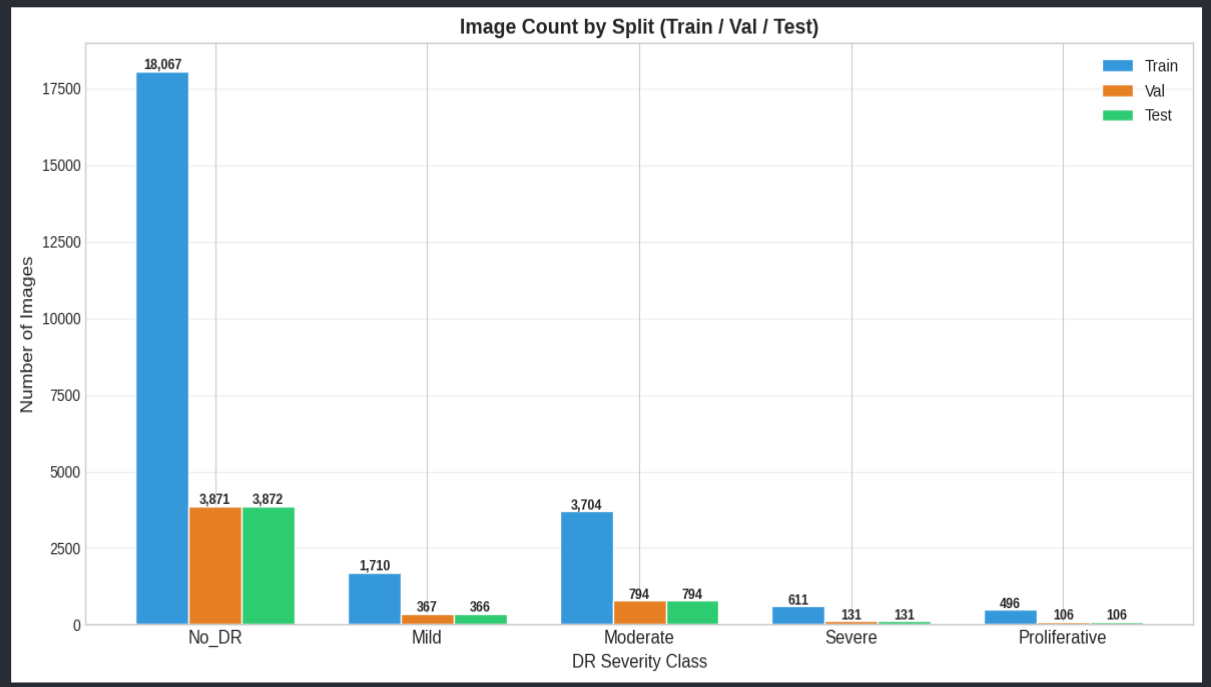
Confusion Matrix (Saved Result)



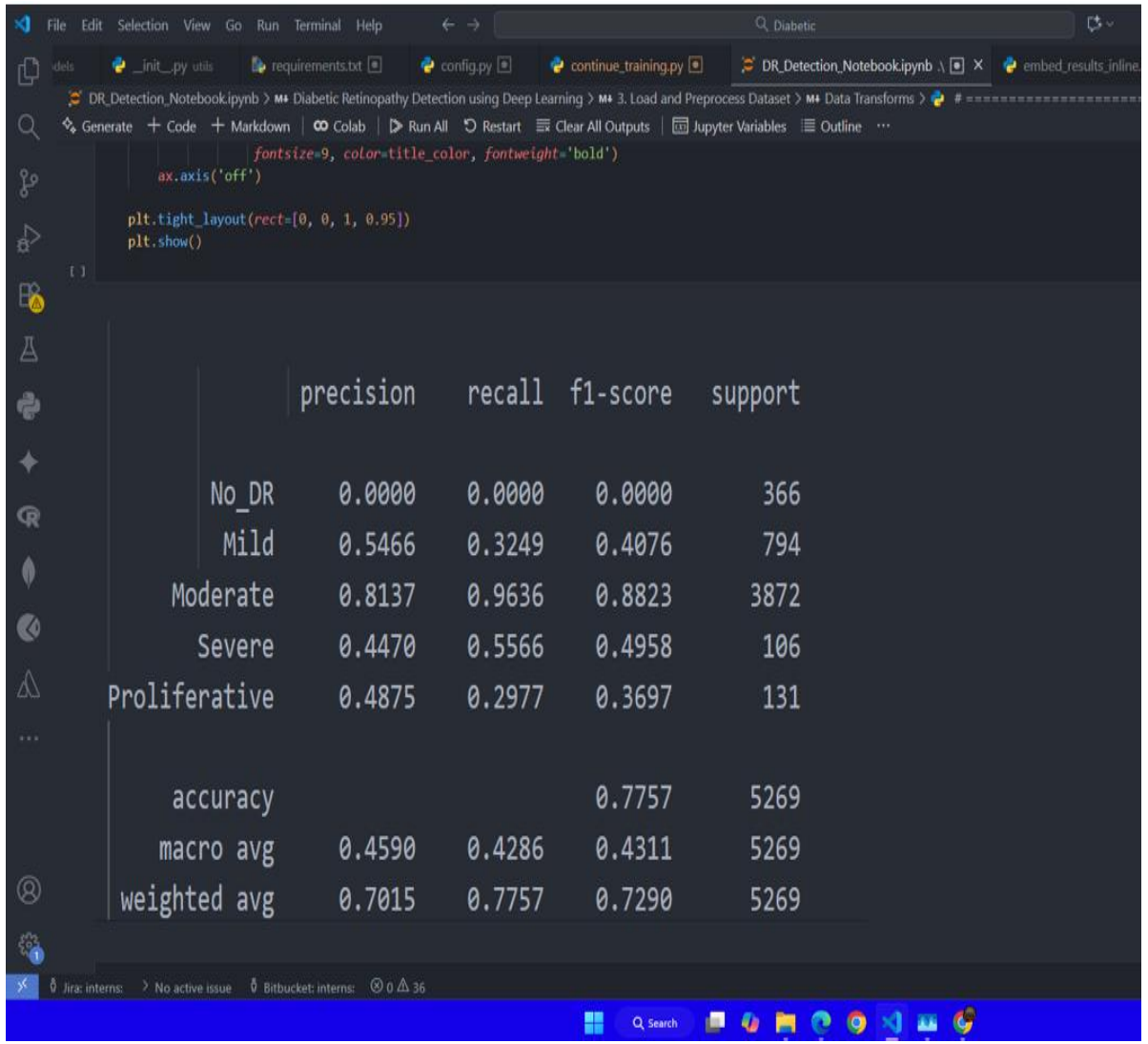
Microsoft Elevate

RESULT:

```
    ax.set_ylabel("Number of Images", fontsize=12)
    ax.set_xlabel("DR Severity Class", fontsize=12)
    ax.set_title("Image Count by Split (Train / Val / Test)", fontsize=14, fontweight='bold')
    ax.legend(fontsize=11)
    ax.grid(True, alpha=0.3, axis='y')
    plt.tight_layout()
    plt.show()
```



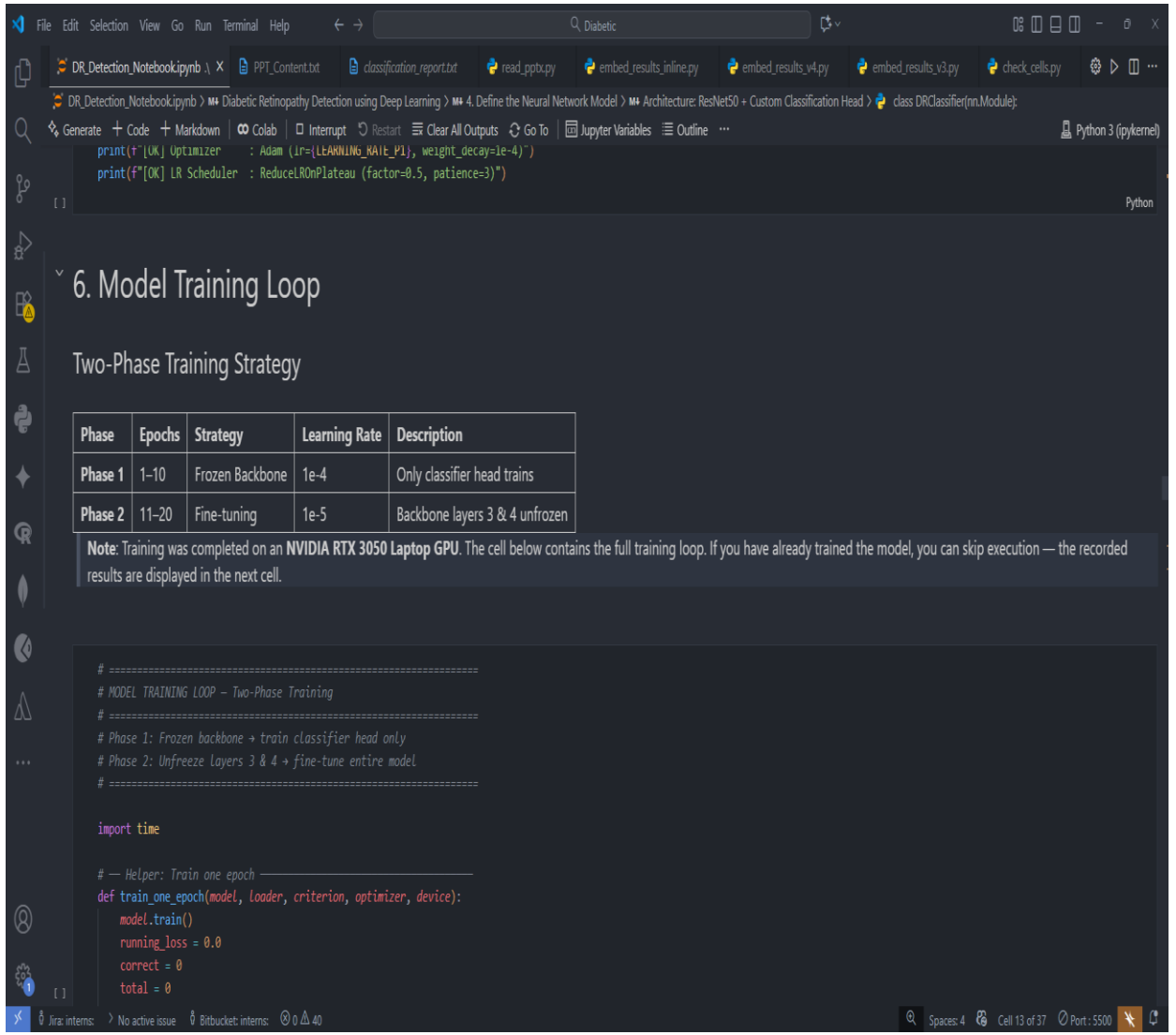
RESULT:



```
                fontsize=9, color=title_color, fontweight='bold')
        ax.axis('off')

    plt.tight_layout(rect=[0, 0, 1, 0.95])
    plt.show()
```

	precision	recall	f1-score	support
No_DR	0.0000	0.0000	0.0000	366
Mild	0.5466	0.3249	0.4076	794
Moderate	0.8137	0.9636	0.8823	3872
Severe	0.4470	0.5566	0.4958	106
Proliferative	0.4875	0.2977	0.3697	131
accuracy			0.7757	5269
macro avg	0.4590	0.4286	0.4311	5269
weighted avg	0.7015	0.7757	0.7290	5269



```
    print(f"[OK] Optimizer     : Adam (lr={LEARNING_RATE_P1}, weight_decay=1e-4)")
    print(f"[OK] LR Scheduler  : ReduceLROnPlateau (factor=0.5, patience=3)")
```

6. Model Training Loop

Two-Phase Training Strategy

Phase	Epochs	Strategy	Learning Rate	Description
Phase 1	1–10	Frozen Backbone	1e-4	Only classifier head trains
Phase 2	11–20	Fine-tuning	1e-5	Backbone layers 3 & 4 unfrozen

> **Note**: Training was completed on an **NVIDIA RTX 3050 Laptop GPU**. The cell below contains the full training loop. If you have already trained the model, you can skip execution — the recorded results are displayed in the next cell.

```
# ============================================================
# MODEL TRAINING LOOP — Two-Phase Training
# ============================================================
# Phase 1: Frozen backbone → train classifier head only
# Phase 2: Unfreeze layers 3 & 4 → fine-tune entire model
# ============================================================

import time

# — Helper: Train one epoch ——————————————————————
def train_one_epoch(model, loader, criterion, optimizer, device):
    model.train()
    running_loss = 0.0
    correct = 0
    total = 0
```

Microsoft Elevate

CONCLUSION:

The Diabetic Retinopathy detection system successfully demonstrates that deep learning with transfer learning can automate retinal image classification into 5 severity levels.- The ResNet50-based model achieved 77.57% overall test accuracy after 20 epochs of two-phase training on an NVIDIA RTX 3060 GPU.

- The two-phase training strategy proved effective:

  Phase 1 (frozen backbone) reached 74.49% validation accuracy

  Phase 2 (fine-tuning) improved it to 76.90%

- The Moderate class showed the strongest performance (96% recall) due to having the most training samples.

- The system highlights the impact of class imbalance -- minority classes (No DR in test set, Severe, Proliferative) had lower recall due to limited training samples.

- This project proves that AI-assisted screening tools can support ophthalmologists in early detection, especially in resource-limited settings where specialist access is scarce.

FUTURE SCOPE:

- **Class Balancing:** Apply Focal Loss, SMOTE oversampling, or class-weighted loss to improve minority class (Severe, Proliferative) performance.
- **Grad-CAM Visualization:** Add explainability by highlighting regions in the retina that influence the model's decision, building trust with clinicians.
- **Higher Resolution Images:** Use 384x384 or 512x512 input size for capturing finer retinal vessel details and microaneurysms.
- **Advanced Architectures:** Experiment with EfficientNet-B4, Vision Transformers (ViT), or ensemble models for higher accuracy.
- **Cloud Deployment:** Deploy on AWS/Azure with a REST API for hospital-level Electronic Health Record (EHR) integration for deployment on mobile devices for rural healthcare workers.
- **Larger Datasets:** Train on the full EyePACS or APTOS dataset(80K+images) for generalization across diverse populations.

Microsoft Elevate

REFERENCES:

1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
2. Gulshan, V., et al. (2016). "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs." JAMA, 316(22), 2402-2410.
3. Pratt, H., et al. (2016). "Convolutional Neural Networks for Diabetic Retinopathy." Procedia Computer Science, 90, 200-205.

Dataset Link: Link

GitHub Link: Link

Microsoft Elevate

Thank You