**Diabetes Detection through Retinopathy (AI Hackathon Report)**

**📝 Introduction**

Diabetic Retinopathy (DR) is a leading global cause of blindness, significantly impacting individuals with diabetes. Early detection is critical to provide effective treatment and avoid serious complications. This project aims to build an accurate and reliable artificial intelligence model that classifies retinal images into five clearly defined severity levels of diabetic retinopathy (DR).

**📚 Dataset Overview**

The dataset contains balanced retinal images classified clearly into five severity levels of diabetic retinopathy:

| **Label** | **Category** | **Description** |
| --- | --- | --- |
| 0 | No DR | Healthy retinal images |
| 1 | Mild DR | Early signs of diabetic retinopathy |
| 2 | Moderate DR | Noticeable retinal damage, moderate severity |
| 3 | Severe DR | Clearly visible, serious damage |
| 4 | Proliferative DR | Most severe, clear abnormalities |

**Format:** JPEG images with structured folder organization (train, val, test) each containing classes labeled (0-4).

**Link to dataset:**  
<https://www.kaggle.com/datasets/kushagratandon12/diabetic-retinopathy-balanced>

**🛠 Methodology**

**✅ Data Preprocessing**

* **Resize images:** 224x224 pixels
* **Normalization:** Standard ImageNet normalization (mean=[0.485,0.456,0.406], std=[0.229,0.224,0.225])
* **Augmentation used:** Horizontal flip, rotation (±15°), brightness and contrast jittering to generalize model effectively.

**✅ Model Architecture**

* **Model used:** EfficientNet-B3 (**pretrained on ImageNet**)
* **Reason for choice:**
  + Efficient and accurate for medical images.
  + Proven good accuracy without excessive computational resources.

**⚙️ Training Details**

* **Loss Function:** CrossEntropyLoss
* **Optimizer:** Adam (Learning Rate = 0.0001)
* **Learning Rate Scheduler:** ReduceLROnPlateau (Patience = 3 epochs)
* **Batch Size:** 64
* **Epochs:** 10 epochs total
* **Early Stopping:** Implemented with patience of 5 epochs to avoid overfitting.

**📊 Results & Evaluation**

**✅ Training and Validation Results**

| **Epoch** | **Training Loss** | **Validation Loss** | **Validation Accuracy** |
| --- | --- | --- | --- |
| 1 | 1.0328 | 0.8800 | 63.45% |
| 2 | 0.7950 | 0.6905 | 70.88% |
| 3 | 0.6563 | 0.6354 | 73.01% |
| 4 | 0.5529 | 0.5341 | 77.34% |
| 5 | 0.4730 | 0.5302 | 78.14% |
| 5 | 0.4047 | 0.4717 | 80.56% |
| 6 | 0.3404 | 0.4660 | 81.56% |
| 7 | 0.2849 | 0.4537 | 82.74% |
| 8 | 0.2453 | 0.4213 | 84.55% |
| 9 | 0.2095 | 0.3904 | 86.04% |

**✅ Final Testing Performance (Unseen Test Data)**

The final model achieved an overall accuracy of **86%** on previously unseen test data. Detailed metrics:

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| No DR | 0.75 | 0.69 | 0.72 | 1000 |
| Mild DR | 0.80 | 0.87 | 0.84 | 971 |
| Moderate DR | 0.78 | 0.77 | 0.77 | 1000 |
| Severe DR | 0.98 | 0.99 | 0.99 | 1000 |
| Proliferative DR | 0.99 | 1.00 | 0.99 | 1000 |

**Overall Accuracy:** **86%**

* Clearly, the model performed exceptionally well for severe (3) and proliferative (4) cases.
* Slightly lower accuracy in early-stage classes (0,1,2), indicating mild/moderate DR stages are harder to distinguish visually.

**🔎 Model Explainability**

**Grad-CAM Visualization** was employed clearly to understand areas critical for the predictions made by EfficientNet-B3.

* **Findings clearly demonstrated:**
  + Predictions heavily rely on **central retinal regions**, aligning well with clinical understanding.
  + Confirms reliability and interpretability of predictions, critical for medical application.

**🎯 Conclusion & Future Improvements**

The EfficientNet-B3 model demonstrated strong performance clearly distinguishing diabetic retinopathy severity (86% accuracy overall). It's especially effective at detecting critical stages of DR (severe/proliferative).

* **Future Recommendations (Clearly Stated):**
  + **EfficientNet-B5 or B7**: Bigger model to improve performance beyond 90% accuracy.
  + **Ensemble Models:** Combine multiple models (CNNs + Transformers) for better accuracy.
  + **Deployment optimization:** Use techniques like quantization, pruning, and TensorRT for faster inference and efficient deployment.

**📋 References**

* EfficientNet Paper: <https://arxiv.org/abs/1905.11946>
* PyTorch EfficientNet: <https://github.com/lukemelas/EfficientNet-PyTorch>
* Grad-CAM: <https://github.com/jacobgil/pytorch-grad-cam>
* Dataset Source: <https://www.kaggle.com/datasets/kushagratandon12/diabetic-retinopathy-balanced>