**Skin Cancer Detection using Deep Learning (CNN, VGG19, ResNet50)**

**🧠 Objective**

To build an automated system using deep learning that accurately detects nine different types of skin cancer. The model is designed to assist dermatologists by improving diagnosis speed and reducing errors, with consideration for diverse skin types and clinical deployment.

**🚨 Problem Statement**

Skin cancer detection currently relies heavily on expert visual analysis, which is subjective and time-consuming. Misdiagnosis or delayed detection can lead to serious consequences. This project focuses on creating a deep learning-based solution that can classify dermatoscopic images of skin lesions into one of nine categories, enabling faster and more consistent detection.

**🔧 Methodology**

* **Dataset Preparation**
  + Labeled skin lesion images were preprocessed and normalized.
  + Data augmentation (rotation, flipping, zooming) was applied to handle class imbalance.
  + Each class was capped at 500 images to avoid model bias.
* **Model Development**
  + Initial implementation used a custom CNN architecture.
  + Due to overfitting, VGG19 and ResNet50 were used through transfer learning.
  + ResNet50 was used as a frozen backbone with additional dense layers for classification.
* **Training & Optimization**
  + Models were trained over 20, 30, and 50 epochs to compare results.
  + Overfitting was addressed using:
    - Dropout Layers
    - L1/L2 Regularization
    - Early Stopping
    - Batch Normalization
    - K-Fold Cross Validation
* **Evaluation Metrics**
  + Accuracy
  + Precision & Recall
  + Confusion Matrix

**📊Results Summary**

| **Model** | **Epochs** | **Training Accuracy** | **Validation Accuracy** |
| --- | --- | --- | --- |
| Custom CNN | 30 | 92% | 69% (Overfitting) |
| VGG19 | 30 | 94% | 75.76% |
| ResNet50 | 50 | 96% | 79% |

🔍 VGG19 helped extract richer features, but ResNet50 performed better by addressing vanishing gradients with its residual connections.

**📸 Visual Results**

Include the following image files in your /images/ folder on GitHub (these were extracted from your PPT):

1. cnn\_architecture.png – custom CNN layers
2. vgg19\_architecture.png – VGG19 model structure
3. resnet50\_architecture.png – ResNet feature map
4. training\_vs\_validation\_accuracy.png – Graph of accuracy/loss vs epochs
5. confusion\_matrix.png – Model prediction comparison
6. sample\_predictions.png – Predicted vs actual outputs

**⚙️ Learning Rate Insight**

The learning rate determines how quickly the model updates weights during training. A high rate might cause the model to overshoot optimal solutions, while a low one leads to very slow convergence. Tuning this parameter was critical to optimizing the performance of VGG19 and ResNet50.

**🔭 Future Work**

* Deploy this model as a web tool for dermatology clinics.
* Improve model interpretability using Grad-CAM visualization.
* Explore ensemble methods or attention-based networks for better accuracy.
* Extend the system to mobile health applications.