Capstone Project 1: Skin Disease Classification using CNN

1. Problem Statement

Develop a deep learning–based **Convolutional Neural Network (CNN)** capable of classifying dermatology images into **20 predefined skin disease categories** using the *20 Skin Diseases Dataset* from Kaggle.

The system supports dermatologists and telemedicine workflows by providing reliable automated predictions.

2. Objective

- Build a complete image classification workflow using CNN.
- Develop robust preprocessing pipelines for medical images.
- Train, validate, and test a multi-class classifier.
- Achieve strong performance using proper evaluation metrics.
- Deliver a clean, reproducible codebase suitable for model deployment (backend only).

3. Dataset Details

Dataset: *20 Skin Diseases Dataset*
Source: Skin Diseases CNN

Dataset Download Code

```
import kagglehub
```

Download the latest version of the dataset

path = kagglehub.dataset_download("haroonalam16/20-skin-diseases-dataset")

print("Path to dataset files:", path)

Dataset Characteristics

- 20 distinct skin disease classes
- Real-world dermatological images
- Variation in lighting, texture, and quality
- No two images are identical → strong generalization required

4. Project Goals

A. Data Preparation

- Load raw images, ensure proper directory structure
- Resize (e.g., 224×224) and normalize images
- Apply augmentations such as:
    - Rotation
    - Flip
    - Zoom
    - Brightness shifts
- Create train/validation/test splits

B. Model Development

Build CNN models using either:

1. **Custom CNN**
    a. Convolution → Activation → Pooling → Dropout → Dense
2. **Transfer Learning - optional**
    a. MobileNetV2

b. EfficientNetB0
c. ResNet50

Both approaches acceptable depending on desired accuracy and compute resources.

C. Model Training

- Train for 20–50 epochs
- Techniques to implement:
  - EarlyStopping
  - ReduceLROnPlateau
  - ModelCheckpoint
- Plot training vs. validation accuracy & loss
- Track overfitting and adjust hyperparameters

D. Model Evaluation

Evaluate using:

- Accuracy
- Precision (per class)
- Recall (per class)
- F1-score (macro & weighted)
- Confusion matrix heatmap
- Classification report
- Error analysis:
  - Misclassified samples
  - Low-confidence predictions

E. Model Saving

- Save trained model in `.h5` (Keras) or `.pt` (PyTorch) format
- Save preprocessing steps

- Save class-label mapping file (`labels.json`)

F. Inference Script (Coding Only, No UI)

A Python script that:

- Loads the trained model
- Preprocesses a given test image
- Outputs predicted class + probability

5. Expected Outcome

A fully-trained CNN model capable of classifying **20 skin disease categories** with high reliability.

This model can be integrated into:

- Teledermatology platforms
- Research workflows
- Backend inference services (no UI needed)
- Clinical decision-support pipelines

6. Final Deliverables Summary

Mandatory

- Data preprocessing pipeline
- Custom or transfer-learning CNN model
- Training script
- Evaluation results + confusion matrix
- Saved trained model
- Documentation (README)