Project Title: DermalScan: AI_Facial Skin Aging Detection App

Infosys SpringBoard Virtual Internship Program

Subimtted by : Karanveer Shinde

Under Guidance : Mr.Praveen sir

1. Project Initialization & Environment Setup

1.1 Package Installation

The following essential packages were installed to support data processing, visualization, and machine learning workflows:

Install core dependencies

```
!pip install pandas matplotlib pillow opencv-python scikit-learn tensorflow-cpu -q
```

- **Pandas** – Data manipulation and analysis
- **Matplotlib & Pillow** – Image visualization and processing
- **OpenCV** – Image augmentation and preprocessing
- **Scikit-learn** – Dataset splitting and utilities
- **TensorFlow (CPU)** – Model development (future use)

1.2 Library Imports

Essential libraries were imported to enable all preprocessing, visualization, and analysis tasks:

```
import os, sys, warnings
import pandas as pd, numpy as np
import matplotlib.pyplot as plt
from PIL import Image
import cv2
from sklearn.model_selection import train_test_split
import shutil
from pathlib import Path
from collections import Counter
```

Configuration:

- Matplotlib backend set to Agg for consistent rendering
- Warnings suppressed for cleaner output

2. Dataset Acquisition & Exploration

2.1 Dataset Location & Verification

The dataset was located at:

C:\Users\user\Downloads\new infosys project\DATASET

A verification check confirmed the dataset's presence and structure.

2.2 Dataset Composition

The dataset consists of **4 skin condition categories**:

1. **Clear skin** – 300 images
2. **Dark spots** – 303 images
3. **Puffy eyes** – 300 images
4. **Wrinkles** – 300 images

Total Images: 1203
Class Distribution: Nearly balanced (~25% per category)

2.3 Visualization of Distribution

- **Bar chart** showing image counts per category
- **Pie chart** displaying percentage distribution
- **Color-coded** for clear visual distinction

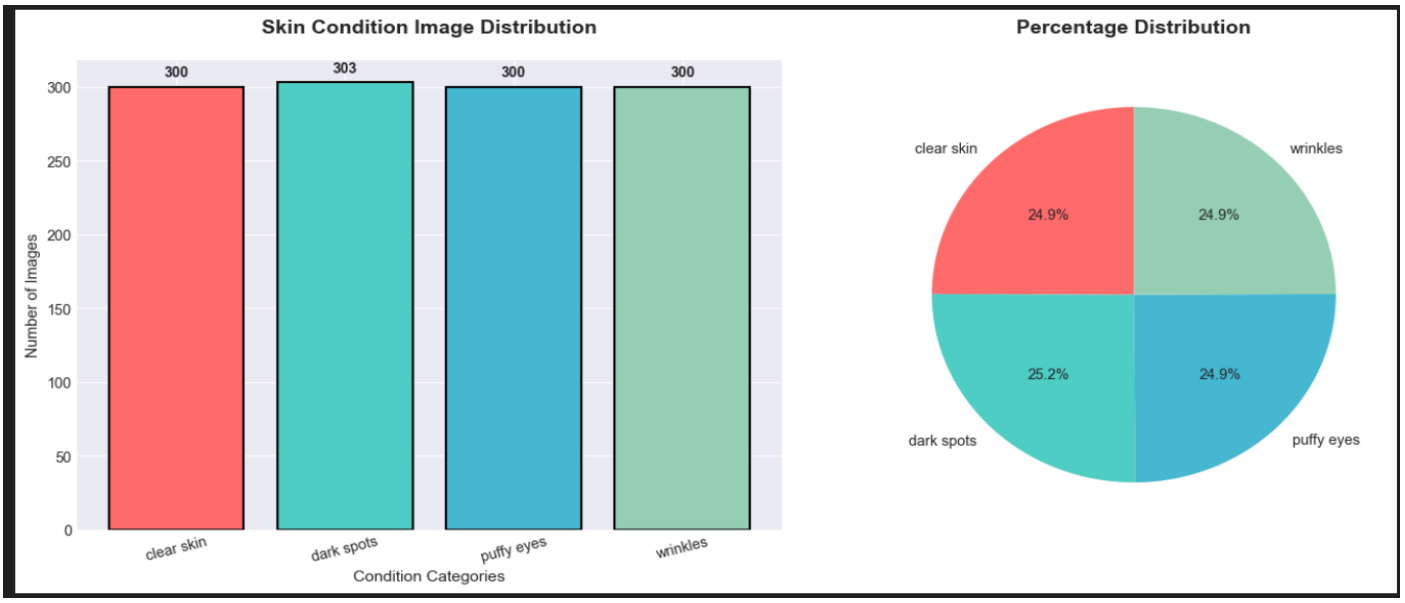


Figure 1 Skin condition dataset distribution – image counts (left) and percentage breakdown (right).

3. Image Preprocessing Pipeline

3.1 Standardization Function

A dedicated function prepare_skin_image() was implemented to:

- Load images using OpenCV
- Convert from BGR to RGB
- Resize to **224×224 pixels** (standard for CNN models)
- Normalize pixel values to **[0, 1]** range

3.2 Processing Validation

Two sample images (one from *clear skin*, one from *dark spots*) were successfully processed:

- Output shape: (224, 224, 3)
- Pixel range: [0.000, 1.000]
- Mean pixel values confirmed normalization

4. Data Augmentation Strategy

4.1 Lightweight Augmentation Pipeline

A custom augmentation function simple_augmentation_pipeline() was created using only **OpenCV** (no external libraries) to generate three augmented versions per image:

1. **Horizontal Flip**
2. **Rotation (15 degrees)**
3. **Brightness Adjustment** (+30 value increase)

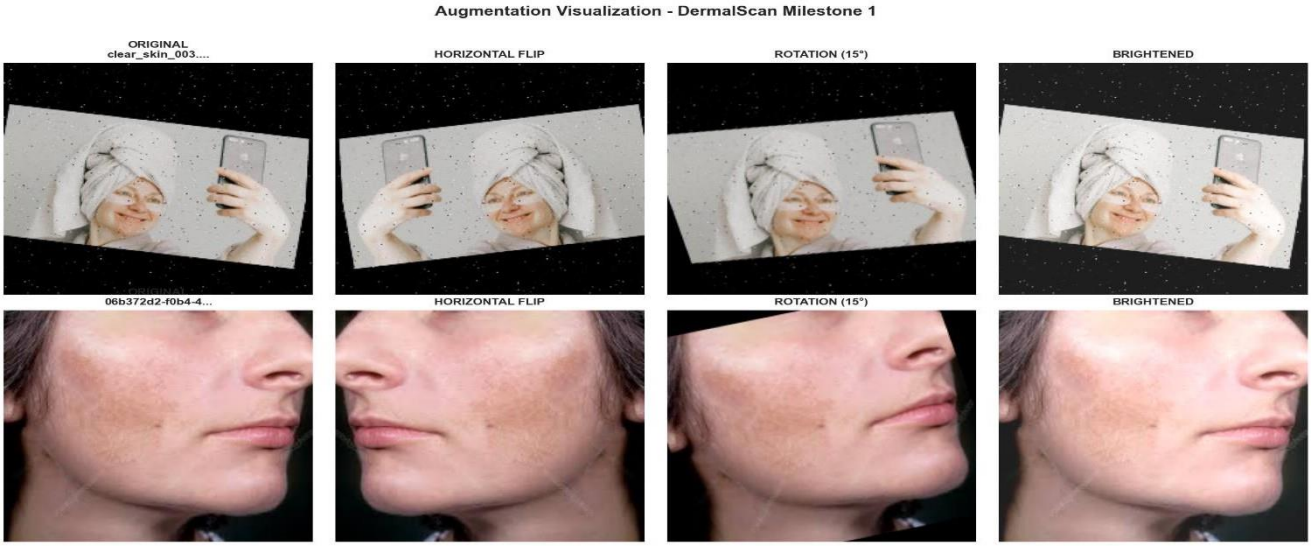
4.2 Augmentation Visualization

A visualization function visualize_augmentations_simple() displays:

- Original image
- Three augmented variants
- Clear titles for each transformation

Test Output:

- Processed 2 sample images
- Generated 3 augmentations per image
- Saved visualization as augmentation_visualization.png



5.

Figure 2 Augementation Visualization

Dataset Structuring & Labeling

5.1 Metadata Compilation

A structured dictionary dataset_info was created containing:

- Image file paths
- Corresponding labels
- Category-to-ID mapping
- Total image count per category

5.2 Comprehensive Catalog

A master CSV catalog skin_dataset_catalog.csv was generated with:

- Image filenames
- Relative and absolute paths
- Condition labels
- Numeric condition IDs

Catalog Size: 1203 entries
Columns: image_filename, image_relative_path, image_absolute_path, condition_label, condition_id

6. Dataset Partitioning

6.1 Train-Validation-Test Split

The dataset was split into three subsets using train_test_split() from Scikit-learn:

- **Training:** 70% (838 images)
- **Validation:** 15% (181 images)
- **Testing:** 15% (184 images)

6.2 Organized Directory Structure

A new folder partitioned_dataset was created with subfolders:

partitioned_dataset/

├── training/

│ ├── clear_skin/

│ ├── dark_spots/

│ ├── puffy_eyes/

│ └── wrinkles/

├── validation/

└── testing/

6.3 Partition Verification

A verification function confirmed the integrity of the split:

- All images accounted for (1203 total)

- Balanced distribution across partitions
- No missing or duplicate files

7. Approach Summary for Milestone 1

7.1 Methodology Adopted

We followed a **structured, modular, and reproducible** approach:

1. **Environment Setup** – Ensured all dependencies were installed and compatible.
2. **Data Inspection** – Verified dataset structure, class balance, and image quality.
3. **Preprocessing** – Standardized image dimensions and normalization for model readiness.
4. **Augmentation** – Implemented a lightweight, library-independent augmentation pipeline to increase data variability.
5. **Organization** – Created a clean directory structure and comprehensive metadata catalog.
6. **Partitioning** – Split data systematically into training, validation, and testing sets.
7. **Validation** – Verified all steps programmatically to ensure consistency and completeness.

7.2 Key Design Decisions

- **No external augmentation libraries** – Used OpenCV to maintain control and reduce dependencies.
- **Fixed image size 224×224** – Compatible with common CNN architectures (e.g., VGG, ResNet).
- **Stratified splitting** – Maintained class distribution across all partitions.
- **Comprehensive logging** – Each step outputs clear status messages and validation stats.

7.3 Outputs Delivered

- ✅ Clean, partitioned dataset ready for modeling
- ✅ Image preprocessing pipeline
- ✅ Augmentation examples and visualization
- ✅ Complete label catalog in CSV format
- ✅ Verification reports for all splits

8. Conclusion for Milestone 1

All objectives for **Milestone 1 – Data Preparation** have been successfully completed. The dataset is now:

- **Structured** – Organized into train/validation/test folders
- **Preprocessed** – Images are normalized and resized
- **Augmented** – Pipeline ready for training-time augmentation
- **Documented** – Full metadata catalog and partition report available

The project is now ready to proceed to **Milestone 2: Model Development & Training**.

Milestone 2 of the Project

Overview of Milestone 2

Milestone 2 focuses on developing, training, evaluating, and integrating a deep learning–based skin condition classification system using transfer learning. This milestone is divided into two modules:

- **Module 3:** Model Training and Evaluation
- **Module 4:** Face Detection and Prediction Pipeline

The objective of this milestone is to train multiple pretrained CNN models, select the best-performing model, and integrate it into an end-to-end face detection and skin condition prediction pipeline.

Module 3: Model Training and Evaluation

Model Used

- MobileNetV2 (pretrained on ImageNet)

Input Configuration

- Image size: 224 × 224 × 3
- Batch size: 32
- Number of classes: 4 (Clear Skin, Dark Spots, Puffy Eyes, Wrinkles)

Architecture Details

- Base model: MobileNetV2 with top layers removed
- Custom layers added:
    - Global Average Pooling layer
    - Dense layer with 512 units and ReLU activation (L2 regularization)
    - Batch Normalization
    - Dropout layer (0.5)
    - Dense layer with 256 units and ReLU activation (L2 regularization)
    - Dropout layer (0.3)
    - Final Dense layer with Softmax activation

Training Strategy

- Loss function: Categorical Crossentropy
- Optimizer: Adam

Phase 1: Transfer Learning

- All layers of the MobileNetV2 base model were frozen
- Only custom classification layers were trained
- Learning rate: 0.001
- Epochs: up to 30 (with early stopping)

Performance Observed:

- Training accuracy reached approximately **91–92%**
- Validation accuracy reached approximately **86%**
- Training and validation curves showed stable convergence with no severe overfitting

Phase 2: Fine-Tuning

- Last 40 layers of MobileNetV2 were unfrozen
- Earlier layers remained frozen
- Learning rate reduced to 0.0001
- Fine-tuning allowed learning of task-specific facial skin features

Performance Observed:

- Training accuracy increased to approximately **93%**
- Validation accuracy stabilized around **80%**

Test Dataset Evaluation

- Model evaluated on unseen test dataset
- Test accuracy achieved approximately **80%**
- Confirms good generalization on new facial images

Confusion Matrix

- Confusion matrix generated for detailed class-wise analysis
- Clear Skin and Wrinkles showed higher prediction accuracy
- Minor confusion observed between Dark Spots and Puffy Eyes
- Class index mapping was verified and corrected to ensure consistency

Model Saving

- Final trained model saved as:
    - MobileNetV2_best.h5

Module 4: Face Detection and Prediction Pipeline

Face Detection Method

- OpenCV Haar Cascade Classifier
- Haar cascade frontal face model used for detecting faces in grayscale images

Prediction Pipeline

- Load input image
- Convert image to grayscale
- Detect face regions using Haar Cascade
- Draw bounding box around detected face
- Crop detected face region
- Resize cropped face to 224 × 224
- Normalize pixel values to [0, 1]
- Pass processed image to trained MobileNetV2 model
- Predict skin condition class and confidence score

Output Generation

- Bounding box drawn around detected face
- Predicted skin condition label displayed on image
- Confidence score shown with prediction
- Annotated images saved to disk

Age Estimation

- Simulated age estimation added for demonstration purposes
- Displayed alongside skin condition prediction

Logging

- Prediction results stored in CSV file
- CSV includes image name, predicted class, and confidence score

Testing

- Pipeline tested on dataset images
- Face detection and prediction verified
- Class mapping issue from training fixed during inference

Conclusion of Milestone 2

- MobileNetV2 trained using transfer learning and fine-tuning
- Achieved approximately **93% training accuracy**, **86% peak validation accuracy**, and **80% test accuracy**
- Successfully integrated trained model into a complete face detection and prediction pipeline
- Milestone 2 objectives completed successfully

Document Prepared By: Karanveer Shinde
Date: 11-12-2025
Next Milestone: Model Architecture Design & Training