INFOSYS SPRINGBOARD VIRTUAL INTERNSHIP 6.0

PROJECT DOCUMENTATION

DERMALSCAN – AI FACIAL SKIN AGING DETECTION APP

Submitted By

G.Likitha

Under the Guidance of

Mr. Praveen

Introduction

Skin aging is a natural process, but early detection of aging signs such as wrinkles, dark spots, puffy eyes, and loss of skin clarity is important for dermatological analysis and cosmetic recommendations. With advancements in Artificial Intelligence and Deep Learning, it has become possible to automatically analyze facial images and classify visible aging features with high accuracy. This aims to build an AI-based Facial Skin Aging Detection System that can recognize and classify different skin aging signs from facial images. Using a combination of image preprocessing techniques, data augmentation, and a deep learning model (EfficientNetB0), the system learns to identify patterns associated with aging. The final output will help detect conditions such as wrinkles, pigmentation, under-eye puffiness, and healthy skin. To develop this system, the initial steps involve preparing a high-quality dataset, organizing it into different categories, and applying preprocessing techniques so the model can learn effectively. The end goal is to create a reliable and user-friendly AI system that can support dermatologists, skincare professionals, and end users by analyzing facial skin images and identifying aging signs accurately.

Problem Statement

Identifying facial skin aging signs such as wrinkles, dark spots, and puffy eyes manually is slow, subjective, and inconsistent. There is a need for an automated system that can accurately analyze facial images and classify these aging signs.
This project aims to develop an AI-based model that can automatically detect and categorize different skin aging features using a labeled dataset, image preprocessing techniques, and a deep learning architecture.

Objectives of the Project

The main objective of the AI DermalScan project is to design and develop a complete end-to-end AI system that can automatically analyze facial images and detect visible signs of skin aging using deep learning and computer vision techniques. The system aims to provide accurate classification of aging indicators—such as wrinkles, dark spots, puffy eyes, and clear skin—and present these results to users in an intuitive and interactive interface.

To achieve this, the project will establish a full pipeline starting from dataset preparation, image preprocessing, augmentation, and model training, followed by face detection, prediction visualization, and deployment through a user-friendly web application. The final goal is to create a reliable, efficient, and real-world applicable tool for skincare assessment, dermatology support, and research purposes.

Milestone 1: Dataset Preparation and Preprocessing

Module 1: Dataset Setup and Image Labeling

1.1 Objective

The goal of Module 1 is to prepare the facial skin–related dataset required for training the skin-aging classification model. This includes collecting images, organizing them into classes, checking data quality, and ensuring proper labeling.

1.2 Dataset Collection

For this project, a facial skin image dataset was used that contains images showing different aging signs.
The dataset mainly contains face images showing skin conditions such as:

- Wrinkles
- Dark Spots
- Puffy Eyes
- Clear Skin

Each image was manually inspected to understand its clarity, relevance, and quality.

1.3 Class Labeling

We created four classes for model training:

Class Name Description

Wrinkles - Visible fine lines and wrinkle patterns

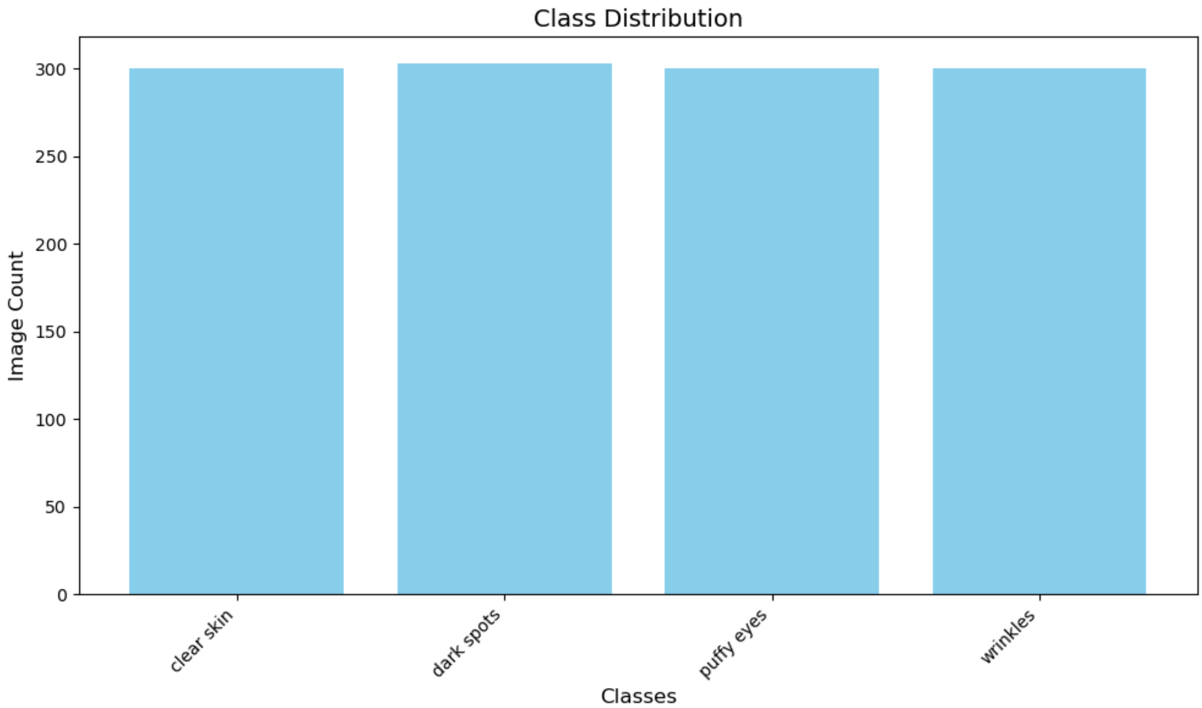
Dark Spots - Pigmented areas or uneven tone

Puffy Eyes - Swollen or sagging under-eye region

Clear Skin - Healthy, smooth skin without major issues

Each image in the dataset was moved into the correct class folder.
Dataset

```
├── wrinkles/
├── dark_spots/
├── puffy_eyes/
└── clear_skin/
```



1.4 Data Inspection

Before preprocessing, the dataset was checked for:

- Blurry or low-quality images
- Incorrectly labeled images
- Duplicate images

Any unclear or unrelated images were removed to maintain dataset quality

Module 2: Image Preprocessing and Augmentation

2.1 Objectives

1. All images resized to 224 × 224 pixels.
2. Pixel values normalized to [0, 1].
3. Augmentation applied to increase data diversity while preserving clinical realism.
4. Example visualizations saved: a 3×3 grid of augmented samples and a before/after pair.

2.2 Image Preprocessing Steps

1. Image Resizing

All images were resized to 224 × 224 pixels, which is the input size required for EfficientNetB0.

```
img = cv2.resize(img, (224, 224))
```

2. Normalization

Pixel values were scaled to the range 0 to 1:

```
img = img / 255.0
```

This helps the model learn faster and reduces training instability.

2.3 Data Augmentation

To increase dataset size and prevent overfitting, augmentation techniques were applied using TensorFlow's ImageDataGenerator.

Augmentation Methods Used

- Flip
- Rotation
- Zoom

These methods create slightly modified versions of original images, improving model generalization.

Original image

Figure 1: *Sample Original Image from the Dataset*

Augmented Images

Figure 2: *Sample Augmented Images*

2.4 Label handling

- Use categorical/one-hot encoding for the four classes.
- Maintain a stable class_indices mapping (e.g., {'clear_skin':0, 'dark_spots':1, 'puffy_eyes':2, 'wrinkles':3})

2.5 One-Hot Encoding

```
y_onehot = to_categorical(y, num_classes=4)
print("One-hot shape:", y_onehot.shape)
```

2.6 Deliverables of Module 2

- Fully preprocessed images (resized + normalized)
- Augmented dataset with increased diversity
- One-hot encoded labels
- Ready-to-use dataset for EfficientNetB0 training