Project Title:DermalScan: AI_Facial Skin Ageing Detection App

Infosys SpringBoard Virtual Internship Program

Submitted By:

Gauri Narlawar to Mr Praveen sir

Project Statement:

The objective is to develop a deep learning-based system that can detect and classify facial aging signs—such as wrinkles, dark spots, puffy eyes, and clear skin—using a pretrained EfficientNetB0 model. The pipeline includes face detection using Haar Cascades, custom preprocessing and data augmentation, and classification with percentage predictions. A web-based frontend will enable users to upload images and visualise ageing signs with annotated bounding boxes and labels.

Outcomes:

- Detect and localise facial features indicating ageing.
- Classify detected features into categories like wrinkles, dark spots, puffy eyes, and clear skin using a trained CNN model.
- Train and evaluate an EfficientNetB0 model for robust classification.
- Build a web-based frontend for uploading facial images and viewing annotated outputs.
- Interate a backend pipeline that processes input images and returns annotated results.
- Export annotated outputs and logs for documentation or analysis.

1. Introduction

Artificial intelligence and deep learning have advanced significantly in the field of computer vision. One emerging application is the automatic detection of facial aging characteristics. Aging indicators such as wrinkles, dark spots, and puffy eyes can provide insights into skin health, lifestyle habits, and dermatological conditions.
This project aims to build a deep learning–based system that automatically identifies facial aging signs from uploaded images. The final system will detect a face, classify visible signs, and present the result through a user-friendly web interface.
Milestone 1 focuses on preparing and preprocessing the dataset, which is a critical step in building a robust and accurate deep learning model.

Objective of Milestone 1

The key objectives of Milestone 1 are:
To collect, organise, and validate the dataset used for training the model.
To ensure images are properly labelled under predefined categories.
To preprocess the images for model readiness, including resizing, normalisation, and augmentation.
To encode labels into a machine-understandable format.
This ensures a strong foundation before moving to model training.

- ***Libraries Used***

  NumPy:

  Purpose: Numerical computing and array manipulation
  Why used: Stores images as arrays (pixel matrices), performs normalisation, reshaping, and efficient mathematical operations.
  Example: converting a list of images into a Numpy array for model training.

- **Panda:**

  Purpose**:** Data tables and label handling
  Why used**:** Helps organise labels and dataset metadata if needed for analysis.

- **OpenCV (cv2):**

  Purpose: Computer vision and image processing
  Role in project:
  Loading images from disk
  Resizing images to 224×224
  Converting colour channels
  Later stages: Face detection using Haar Cascade

- **Matplotlib & Seaborn**

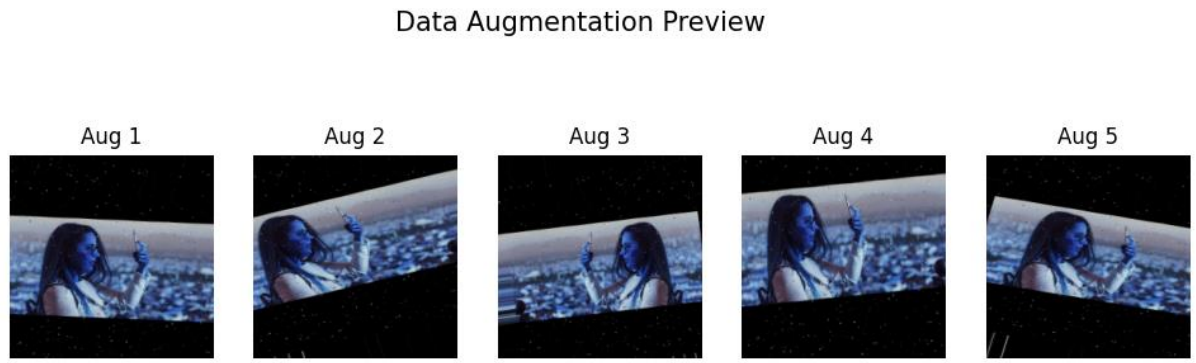
Purpose: Data visualisation and plotting
Used for:
Class distribution bar chart
Image visualization
Understanding dataset balance

- **Scikit-Learn (sklearn)**
  Purpose: Machine learning utilities
  Used for:
  Train-test split
  Label encoding
  Evaluation metrics

- **TensorFlow / Keras**
  Purpose: Deep learning framework
  Role in project:
  Image augmentation
  One-hot encoding of labels
  Later: EfficientNetB0 model training

Code:

```
import numpy as np
import random
import matplotlib.pyplot as plt
random_index = random.randrange(len(processed_images))
selected_image = processed_images[random_index]
image_array = np.expand_dims(selected_image, axis=0)
augmented_batch = [next(datagen.flow(image_array))[0] for _ in range(5)]
plt.figure(figsize=(12, 4))
for i, img in enumerate(augmented_batch):
    plt.subplot(1, 5, i+1)
    plt.imshow(img)
    plt.axis("off")
    plt.title(f"Aug {i+1}")
plt.suptitle("Data Augmentation Preview", fontsize=15)
plt.show()
```



Conclusion of Milestone 1:

Milestone 1 successfully completes the dataset preparation and preprocessing stage. The dataset is now organised, cleaned, normalised, encoded, and augmented. This ensures that the upcoming model training phase (Milestone 2) can proceed efficiently with high-quality input data.