DermalScan – AI Facial Skin Aging Detection App

Infosys Springboard Virtual Internship Program

Submitted by,

Adithya Krishna

Under the Guidance of

Mr. Praveen

Mentor, Infosys Springboard

Introduction

With the rapid advancement of computer vision and deep learning, automated facial analysis has become a significant area of research. Age estimation from facial images plays a crucial role in various domains including dermatology, skincare, security, personalized recommendations, and digital health applications. Traditional methods of assessing facial age often rely on manual observation, which can be subjective, inconsistent, and time-consuming.

To address these limitations, this project proposes an **AI-based Facial Age Detection System** capable of predicting the age category of an individual from a facial image. Using convolutional neural networks (CNNs), data augmentation, and a well-structured dataset, the system learns visually identifiable age-related features such as wrinkles, skin texture, and facial structure changes.

This project aims to deliver an efficient, scalable, and accurate model that supports real-world usage and showcases the practical application of deep learning in facial analysis. The developed system serves as a valuable contribution to the growing field of AI-driven dermatological and biometric solutions.

Problem Statement

This project will provide a fast, reliable, and scalable solution that can support dermatologists, skincare brands, and users by delivering objective age predictions with minimal human intervention. A web-based frontend will enable users to upload images and visualize aging signs with annotated bounding boxes and labels.

Objectives of the Project

- To design an AI system capable of detecting and localizing key facial features associated with aging.
- To classify age-related facial characteristics—such as wrinkles, dark spots, puffy eyes, and clear skin—using a trained Convolutional Neural Network (CNN).
- To train and evaluate a deep learning model to achieve robust and accurate facial feature classification.
- To develop an interactive web-based frontend that allows users to upload facial images and view the annotated results.
- To implement a backend processing pipeline that analyzes input images and returns predictions along with visual annotations.
- To enable export of annotated outputs and system logs for documentation, reporting, or further analysis.

Dataset Preparation and Preprocessing (Milestone-1)

1. Introduction

The objective of Milestone 1 is to prepare a clean, labelled, balanced, and preprocessed dataset for training a model that detects facial age-related skin features such as wrinkles, dark spots, puffy eyes, and clear skin.
This milestone ensures high-quality data for efficient model training in Milestone 2.

2. Dataset Setup

2.1 Folder Structure

```
project_root/
├── dataset/
├── cleaneddataset/
├── splittedataset/
│   ├── train/
│   └── validation/
│   └── test/
```

dataset – Main directory containing all project image data, including the cleaned dataset and the train/validation split.

cleaneddataset – Contains cleaned and preprocessed images after removing low-quality files, duplicates, and incorrectly labelled examples. This acts as the master dataset before splitting.

splittedset – Contains the dataset split into train/ and validation/ folders for model training and performance evaluation. Generated from the cleaned dataset.

notebooks – Contains Jupyter notebooks for data analysis, visualization, preprocessing, and model experimentation.

2.2 Class Definitions

Images were categorized into the following four classes:

- Clear Skin
- Wrinkles
- Dark Spots
- Puffy Eyes

Ambiguous or low-quality images were removed to maintain dataset purity.

3. Dataset Inspection and Class Distribution

A short script was used to verify:

- Images are correctly labelled
- Samples display properly

```
Import os
for c in classes:
    clfolder = os.path.join(datasetpath,c)
    imagename = os.listdir(clfolder)[0]
    imgagepath = os.path.join(clfolder,imagename)
    img = cv2.imread(imgagepath)
```

3.1 Class Distribution Plot

To check dataset balance, the number of images per class was counted and plotted.

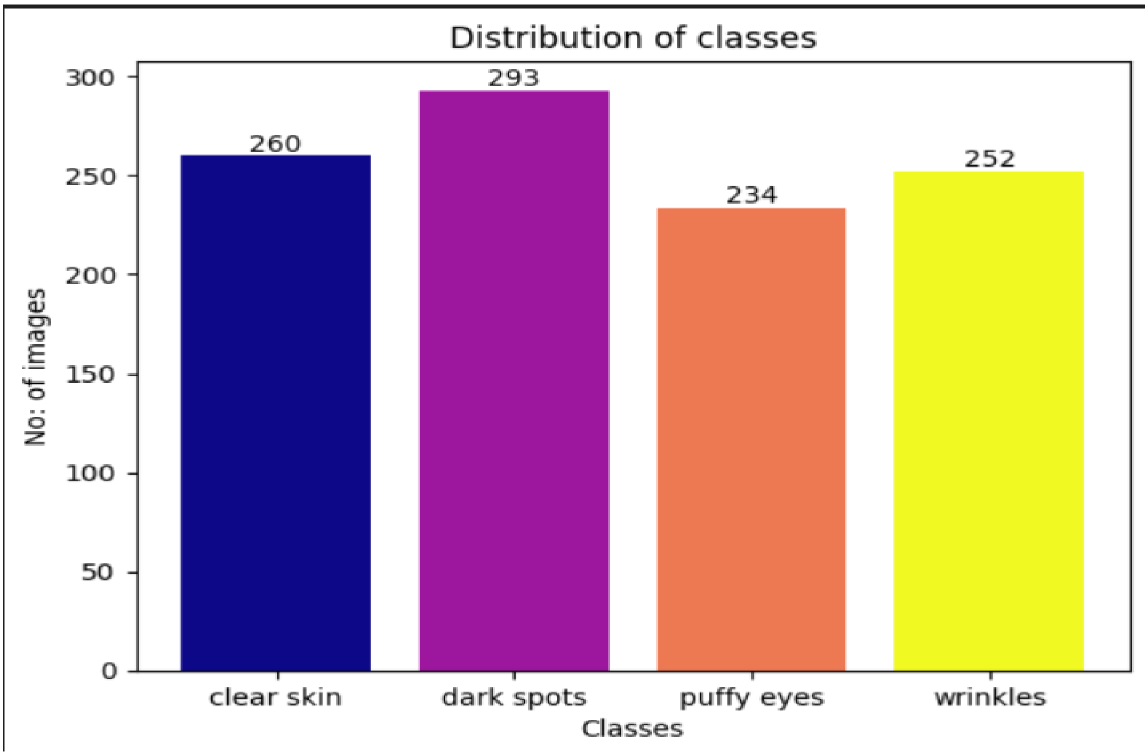

Figure 1:*Distribution of images across 4 classes*

4. Test-Train-Validation Split

The dataset was divided into three subsets using a 70:20:10 ratio, where 70% of the images were allocated for training, 20% for validation, and 10% for testing. All three subsets contain the same four classes: clear skin, wrinkles, dark spots, and puffy eyes.

- **Training Set (70%)**
  Used to fit the model and learn the distinguishing features of each class.
- **Validation Set (20%)**
  Used during training to tune hyperparameters and evaluate model performance on data the model has not seen before.
- **Test Set (10%)**
  Used only after training is complete to measure the model's final performance and generalization ability on completely unseen data.

This three-way splitting strategy ensures that the model is trained effectively, validated during development, and rigorously evaluated after training, thereby improving its reliability and real-world applicability.

5. Image Preprocessing

5.1 Resize and Normalize

All images were resized to **224 × 224 pixels** for model compatibility. Pixel values were normalized by dividing each value by **255**, ensuring all inputs fall within the 0–1 range.

5.2 One-Hot Encoding

Labels were automatically converted into one hot encoding using class_mode="categorical",

Allowing the model to handle multi class classification.

6. Data Augmentation Pipeline

Augmentation was applied only to training images to increase diversity while preserving class features. These transformations simulate natural changes that occur in facial photos without altering the underlying age-related features.

Transformations Used

- **Rotation (±20°)** – Simulates slight head tilts, helping the model remain accurate even when faces are not perfectly aligned.
- **Zoom (up to 20%)** – Mimics variations in camera distance, ensuring the model can detect age features such as wrinkles or dark spots even when the face appears closer or farther away.
- **Horizontal/Vertical Shifts (10%)** – Represents small changes in face positioning within the frame, preventing the model from becoming sensitive to fixed face alignment.

- **Horizontal Flip –** Helps the model generalize to left/right orientation of facial features, since aging signs appear symmetrically.
- **Brightness Variation –** Accounts for different lighting conditions, ensuring age-related features are recognized in bright or dim environments.
- **Fill mode: Nearest –** Used to fill in newly created pixels during transformations while preserving important facial details.

These augmentations improve robustness by teaching the model to focus on age-indicative patterns rather than on fixed image positions, lighting, or orientation.

```
traingenerator = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    brightness_range=(0.9, 1.1),
    fill_mode="nearest")
```

7. Visualization of Output

7.1 Original Images

These images represent some sample images from the dataset before augmentation.

Figure 2*: Sample Original Images from the Dataset*

7.2 Augmented Images

These images demonstrate how augmentation transforms some sample images from the dataset.

***Figure 3**: Sample Augmented Images from the Dataset*

8. Conclusion

Milestone 1 successfully produced a clean, structured, and enhanced dataset ready for model development. All required inspection, preprocessing, and augmentation tasks were completed. This forms the foundation for next module.