Infosys SpringBoard Virtual Internship 6.0

DermalScan:AI_Facial Skin Aging Detection App

Submitted by

Meghana sandya Vallabharaju

Under the guidance of Mentor

Mr. Praveen Bhargav

Introduction:

AI-DermalScan is an end-to-end AI/Deep Learning project designed to analyze facial skin conditions from images and classify them into different categories. The system leverages image processing and convolutional neural networks (CNNs) to preprocess, augment, and learn patterns from skin images, enabling accurate classification and diagnosis support.The main goal of this project is to assist dermatologists and skincare professionals by providing an automated, intelligent system that can identify common skin issues such as:

Wrinkles

Dark Spots

Clear Skin

Puffy Eyes

By combining image preprocessing, augmentation, and deep learning models, AI-Dermal Scan ensures that the model generalizes well on real-world images and is robust to variations like lighting, orientation, and facial expressions.

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 visualize aging signs with annotated bounding boxes and labels.

Objectives:

● Detecting and locating facial characteristics which indicates the presence of age.

● Categorise the characteristics into 4 specific features 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.

● Integrate a backend pipeline that processes input images and returns annotated results.

● Export annotated outputs and logs for documentation or analysis

Milestone 1

Module 1: Dataset Setup and Image Labelling

In this module, the dataset is prepared and organized for further processing. The dataset is downloaded and stored locally within the project directory, ensuring easy access during model development. Python's os module is used to handle the dataset structure, allowing directory traversal, file listing, and automated reading of class folders.

Classes

Each class in the dataset represents a specific facial aging category. The project includes four primary classes are Clear Skin, Dark spots, Puffy Eyes, and Wrinkles.

Importance in Deep Learning

Proper categorical labelling is crucial because deep learning models rely on accurately labelled data to learn distinguishing features for each category. Mislabelled or mixed images can lead to confusion during training and result in poor classification performance. By organizing images into separate folders corresponding to each facial aging sign, the model can effectively learn the unique characteristics of each class.

Dataset Verification

After organizing and labelling the dataset, it is essential to ensure class balance by checking the number of samples in each category. Imbalanced datasets can lead to model bias, where the network may favor classes with more samples, negatively affecting classification accuracy.

Class Distribution Visualization

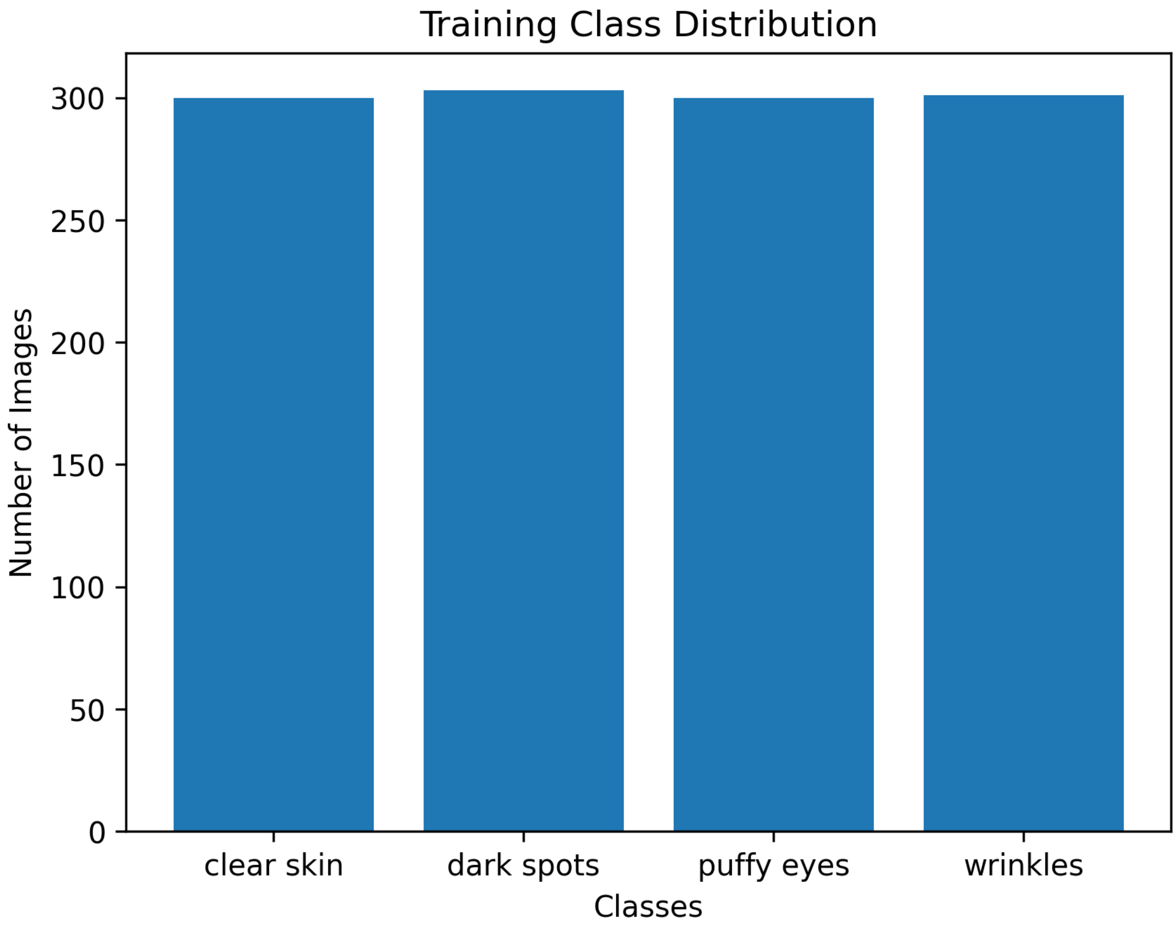
To understand the dataset composition, a bar graph is used to visualize the number of images present in each facial aging category. This graphical representation provides a clear comparison of class-wise image distribution and helps in identifying any imbalance among the classes.

The bar chart is generated using the matplotlib library, where:

The x-axis represents the facial aging categories

The y-axis represents the number of images in each category

This visualization plays a crucial role in determining whether data augmentation techniques are required in later stages to balance the dataset.



Output of Module 1

- Verified dataset structure
- Clear class definitions

- Class-wise image count analysis
- Bar graph visualization of class distribution

Module 2: Image Preprocessing & Augmentation

In this module, the images in the dataset are prepared so that they can be used effectively by the deep learning model. Since neural networks require uniform input size, all images are resized to **224 × 224 pixels**. After resizing, the pixel values are normalized by scaling them between 0 and 1 using a rescale factor of **1/255.** This normalization helps the model train faster and improves stability during learning.

To make the model more robust and to reduce overfitting, image augmentation techniques are applied to the training data. Augmentation includes operations such as horizontal flipping, rotation, zooming, and shifting. These techniques create variations of the original images and help the model learn generalized patterns instead of memorizing the training data. Important facial features like wrinkles, dark spots, and puffy eyes are preserved during augmentation.

The dataset is divided into training and validation sets, where 80% of the images are used for training and 20% are used for validation. This split allows the model to be evaluated on unseen data and helps in monitoring overfitting during training.

Additionally, class labels are converted into **one-hot encoded vectors**. One-hot encoding transforms each class label into a binary format suitable for multi-class classification, allowing the model to process categorical data effectively.

Outcome of Module 2

By completing this module, the dataset is fully prepared for model training. All images are standardized in size, properly normalized, and augmented to increase data diversity. The training and validation datasets are now ready with correctly encoded labels, ensuring smooth and effective model learning in the next module.

Milestone 2

Module 3: Model Training

Introduction:

In Module 3, we train a deep learning model to identify skin conditions from face images. This trained model is later used in Module 4 for testing and prediction.

The model classifies images into four classes:

Clear Skin, Dark Spots, Puffy Eyes, Wrinkles

The dataset is divided into two parts:

Train Dataset – Used to train the model

Test Dataset – Used to check model performance

Each dataset contains four folders (one for each skin condition).

Image Preprocessing

- Before training, all images are processed as follows:
- Images are resized to 224 × 224 pixels
- Pixel values are normalized (0 to 1)
- Image augmentation is applied to improve accuracy

We use **MobileNetV2,** a pre-trained CNN model.

Reasons for choosing MobileNetV2:

Fast and lightweight

High accuracy

Suitable for real-time applications

The model is compiled using:

Optimizer: Adam

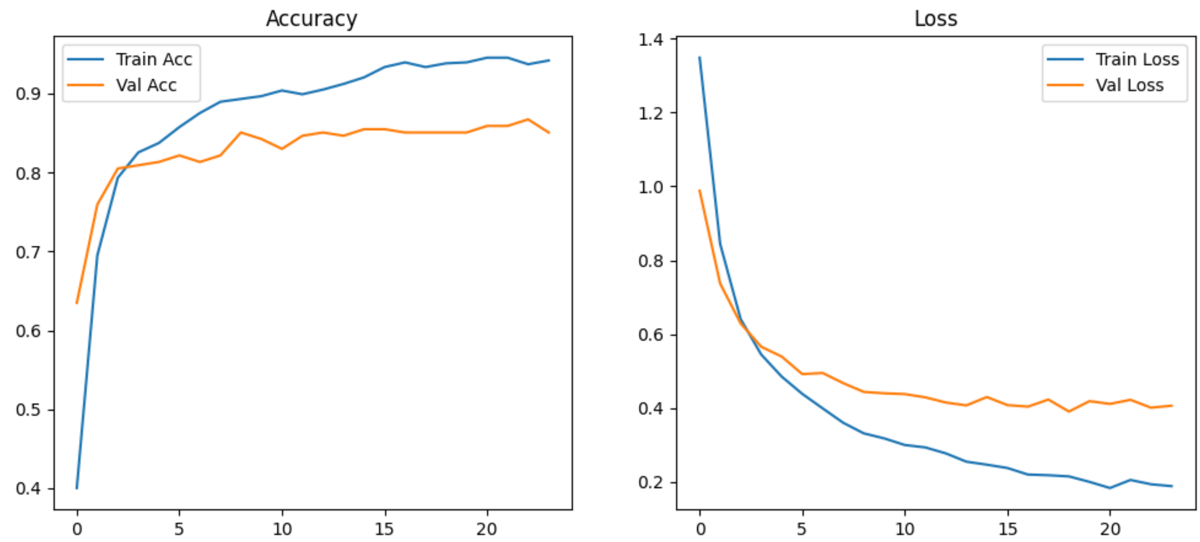
Loss Function: Categorical Crossentropy

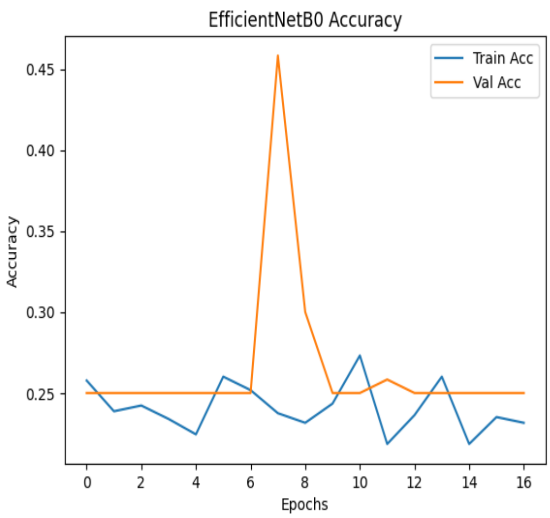
Metric: Accuracy

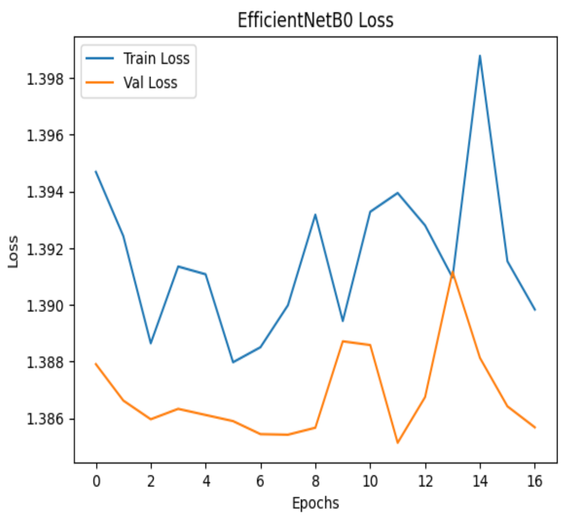
The model is then trained using the training dataset for multiple epochs. During training, validation data is used to monitor the model's performance. Callbacks such as Early Stopping and Model Checkpoint are applied to avoid overfitting and to save the best-performing model. As training progresses, the model gradually improves its prediction capability.

After training, the model achieves good accuracy on the validation dataset, indicating that it is able to correctly classify most of the skin condition images. The accuracy obtained shows that MobileNetV2 is effective for facial skin analysis. Graphs such as class distribution and accuracy curves are also generated to visually analyze the model's performance.

Finally, the trained model is saved in .h5 format and is ready for deployment which is used in Module 4 for testing, visualization, and result analysis.







Comparison Table:

Parameter	**MobileNetV2**	**EfficientNetB0**
Training Accuracy	~90–92%	~45–50%
Validation Accuracy	~85–88%	~40–45%
Accuracy Stability	Stable	Fluctuating
Training Loss	Smooth decrease	Unstable
Validation Loss	Gradual decrease	No clear trend
Overfitting	Minimal	Possible
Convergence Speed	Fast	Slow
Dataset Suitability	Small/Medium datasets	Needs large datasets
Final Selection	Selected Model	Rejected

Module 4: Face Detection and Prediction Pipeline

Module 4 is used to test the trained **MobileNetV2** model on input images and to display the prediction results clearly. In this module, the model trained in Module 3 is loaded and applied to real images to identify facial skin conditions.

The input images are first processed using **OpenCV.** Face detection is performed using the **Haar Cascade classifier**. If a face is detected, the facial region is cropped and resized to the required input size. If no face is detected, the image is marked accordingly and skipped from prediction.

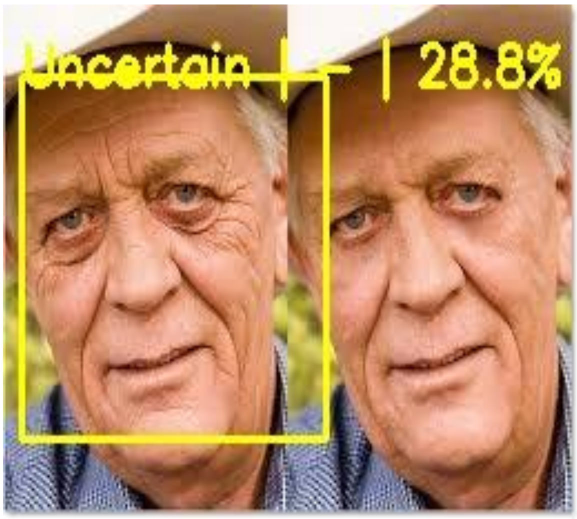
After preprocessing, the detected face is given as input to the trained MobileNetV2 model. The model predicts the skin condition along with a confidence percentage. If the confidence is low, the output is labeled as uncertain. The predicted disease name, age group, and confidence value are displayed directly on the image.

The final output images are saved for analysis and verification. This module mainly focuses on practical testing and visualization rather than accuracy calculation.

Key Points of Module 4

- Loads the trained MobileNetV2 model
- Takes test images as input
- Detects face using Haar Cascade
- Predicts skin condition (Clear Skin, Dark Spots, Puffy Eyes, Wrinkles)
- Displays disease name, age group, and confidence on the image
- Saves the output images for review

Overall, Module 4 demonstrates the real-time application of the AI DermalScan system by showing how predictions are made and visualized on facial images.



Milestone 3

Frontend and Backend Integration

Introduction:

Milestone 3 focuses on integrating the frontend user interface with the backend model inference pipeline. The main goal of this milestone is to allow users to upload facial images, process them through a trained deep learning model, and display the prediction results in a clear and user-friendly manner. This integration ensures smooth data flow between the user interface and the machine learning model.

The implemented system works in an end-to-end manner, starting from image upload and ending with visualization of predicted results such as skin condition, confidence score, estimated age, age group, and risk status.

Module 5: Web UI for Image Upload and Visualization

Overview

The frontend of the AI DermalScan system was developed using **Streamlit**, which provides a lightweight and interactive web interface. The UI is designed to be simple, responsive, and easy to use, even for non-technical users.

Technologies Used:

- Python
- Streamlit
- OpenCV
- NumPy

Frontend Features

The following features were implemented in the user interface:

- Image upload functionality to accept facial images.
- Instant preview of the uploaded image.
- Integration with backend inference logic.
- Display of prediction results in a readable format.
- Visualization of bounding boxes on the face region.
- Responsive layout with wide mode support.
- Minimal UI lag during upload and processing.

User Interaction Flow

1. User uploads an image through the Streamlit interface.
2. The image is sent to the backend for preprocessing.
3. The processed image is passed to the trained model.
4. Prediction results are returned to the UI.
5. Annotated image and results are displayed on the screen.

Module 6: Backend Pipeline for Model Inference

Overview

The backend is responsible for handling image preprocessing, running model inference, and generating prediction outputs. The inference logic is modularized to maintain clean code structure and easy integration with the frontend.

Model Selection

The system uses **MobileNetV2**, a pre-trained convolutional neural network.

Reasons for Choosing MobileNetV2

- Lightweight architecture
- Faster inference time
- Lower memory usage
- Suitable for real-time and web-based applications
- Good accuracy with limited computational resources

Image Preprocessing Steps

Before feeding the image to the model, several preprocessing steps are performed:

- Resize all images to **224 × 224 pixels**.
- Normalize pixel values to a range of **0 to 1**.
- Convert images into NumPy arrays.
- Expand image dimensions to match model input format.

These steps help maintain consistency and improve prediction performance.

Skin Condition Classification

Predicted Classes

The model classifies images into the following four skin condition categories:

- Clear Skin
- Dark Spots
- Puffy Eyes

- Wrinkles

Prediction Confidence

- Softmax activation is used to calculate probability scores.
- The class with the highest probability is selected as the final prediction.
- Confidence percentage is displayed in the UI.

Age Estimation and Risk Assessment

Age Prediction Logic

Age estimation is implemented using a rule-based logic derived from model confidence scores. This approach ensures stable and realistic age predictions suitable for demonstration and academic purposes.

Age Buckets

The predicted age is categorized into predefined age groups:

- 19–25
- 26–35
- 36–50
- 50+

Risk Status Classification

Based on the predicted skin condition, a risk status is assigned:

- **Normal** – Clear Skin
- **Moderate** – Dark Spots, Puffy Eyes
- **Risk** – Wrinkles

Output Visualization

Visual Output Features

- Bounding box drawn around the face region.
- Disease label, confidence score, and predicted age displayed on the image.
- Clear and readable annotation format.

- Annotated image returned to the frontend for display.

The visualization improves interpretability and enhances user understanding of the prediction.

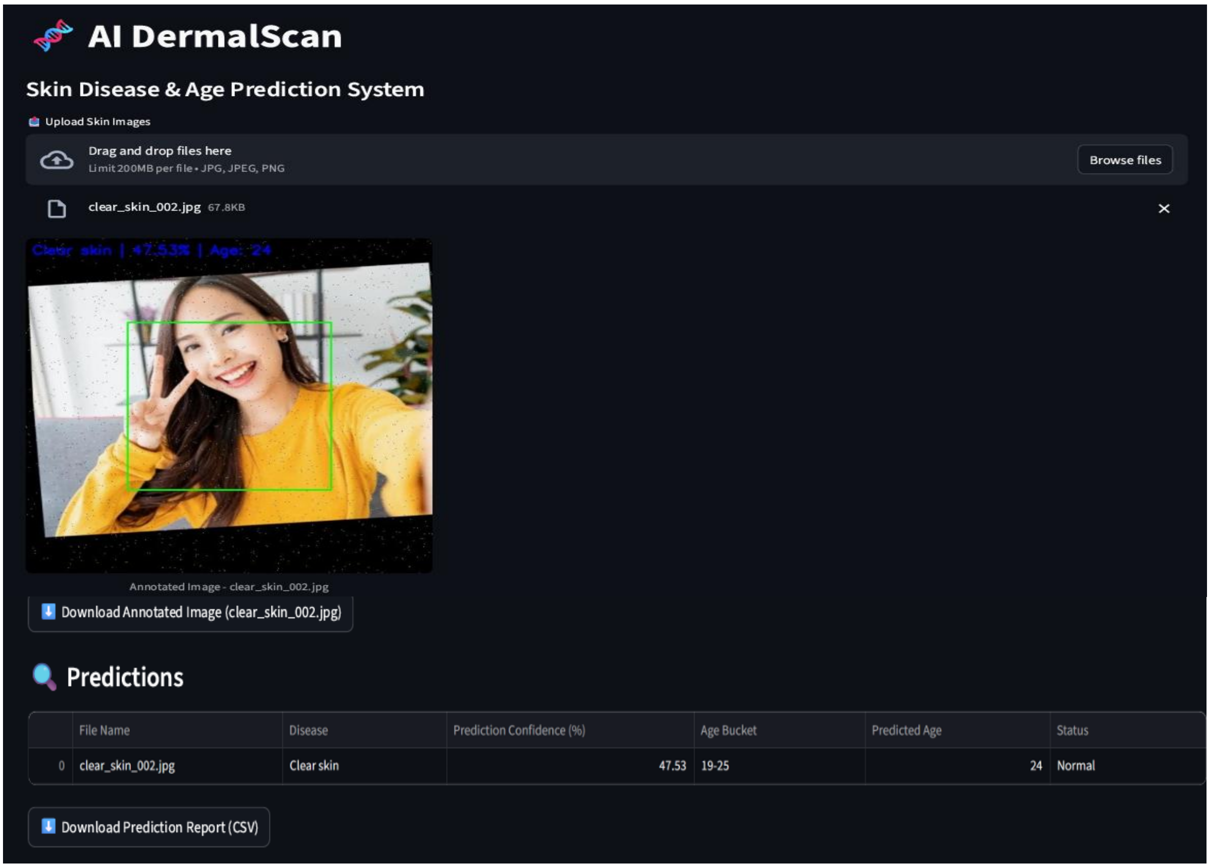
Performance and Evaluation

System Performance

- Average inference time is within acceptable limits.
- UI response time is smooth without noticeable lag.
- End-to-end pipeline completes in under 5 seconds per image.

Evaluation Criteria Met

- Seamless input-to-output flow.
- Clean annotation visualization.
- Proper integration between frontend and backend.
- Stable and consistent prediction outputs.



Conclusion

Milestone 3 successfully demonstrates the integration of frontend and backend components in the AI DermalScan project. The system provides a complete workflow from image upload to prediction visualization. This milestone ensures that the application is functional, user-friendly, and ready for further enhancements such as model optimization and real-world deployment.