DermalScan:AI_Facial Skin Aging Detection App

INFOSYS SPRINGBOARD VIRTUAL INTERNSHIP 6.0

BATCH - 9

Submitted by:

Farhan Akhtar

Under the guidance of Mentor ***Mr. Praveen***

Milestone 1: Dataset Preparation and Preprocessing –

(MODULE 1)

➢ INTRODUCTION –

- Facial skin aging is one of the most common concerns related to personal health and appearance. With the growing advancements in artificial intelligence, deep learning has become a powerful solution to analyze facial features and detect aging signs more accurately than traditional manual observation.
  The goal of this project — DermalScan — is to build an AI-based system that can identify different signs of facial aging such as wrinkles, dark spots, puffy eyes, and clear skin. The application uses a pre-trained EfficientNetB0 model along with face detection techniques, preprocessing, and a prediction pipeline to provide results that are reliable, fast, and user-friendly.
- This project is not just for automated predictions; it also focuses on creating an interface where users can upload their photos and receive annotated visual results with predicted percentages. In brief, DermalScan aims to assist in skin analysis using artificial intelligence in an accessible and practical way.

➢ PROJECT OBJECTIVE –

The main objective of DermalScan is:

- To detect facial features that indicate different signs of aging and classify them into four categories: wrinkles, dark spots, puffy eyes, and clear skin.
- To provide visual outputs such as annotated images and prediction percentages that users can easily understand.
- To build a complete pipeline from dataset preparation to frontend deployment.

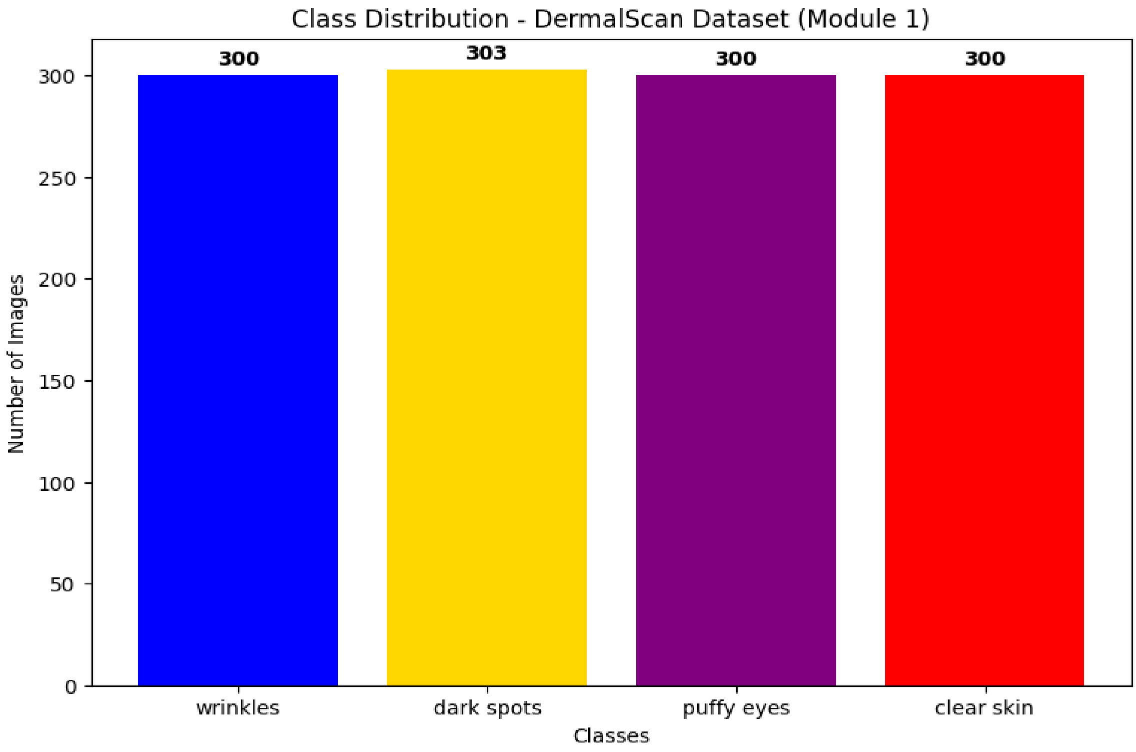
➢ Module 1 — Dataset Setup & Image Labelling –

Module 1 was focused on preparing the dataset and ensuring that the model has high-quality images for training. According to the project structure, this module includes:

- Collecting/setting up a dataset of facial images
- Categorizing them into four classes: wrinkles, dark spots, puffy eyes, clear skin
- Making sure each class has enough samples and the distribution is balanced

• Bar Graph — Class Distribution

A bar graph of the dataset was generated using the dataset inspection notebook to show the **number of images per class**. This visual output helped confirm that no class was significantly over- or under-represented.



➢ Final Output of Module 1

- Dataset fully sorted and labeled
- Class folders prepared for preprocessing
- Graph confirms balanced dataset
- Ready to proceed to Module 2

(MODULE 2)

INTRODUCTION -

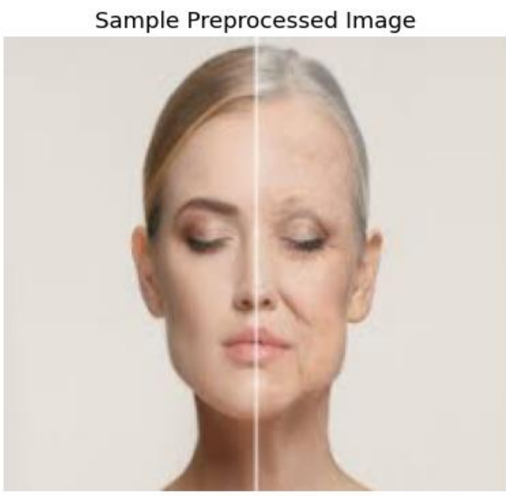
In this module, the dataset prepared in Module 1 is processed to make it suitable for deep learning. The main focus is resizing, normalization, augmentation, and label encoding.

➢ Objective -

- Prepare images for CNN input
- Increase dataset diversity using augmentation
- Convert labels into machine-readable format

➢ Preprocessing Steps -

- Images resized to 224×224
- Pixel values normalized to 0–1
- Images converted to NumPy arrays



➢ Data Augmentation -

To improve model generalization, basic augmentation techniques were applied:

- Horizontal flipping
- Rotation
- Zooming

➢ Label Encoding -

- Wrinkles → 0
- Dark Spots → 1
- Puffy Eyes → 2
- Clear Skin → 3
  Labels were later converted to one-hot encoded vectors.

➢ Output of Module 2 -

- Preprocessed image dataset
- Augmented samples for training
- One-hot encoded labels
- Dataset ready for CNN model training

➢ Conclusion -

- Module 2 successfully prepares the DermalScan dataset for deep learning by applying preprocessing and augmentation techniques, ensuring improved model performance in the next stage.

Milestone 2: Model Training and Evaluation -

(MODULE 3)

Objective

The primary objective of **Module 3** is to convert the preprocessed and augmented images into a structured, machine-learning ready dataset. While Modules 1 and 2 focused on understanding the dataset and enhancing it through preprocessing and augmentation, this module ensures the data is transformed into a format suitable for deep learning model training. This stage establishes a vital connection between raw image data and the model training pipeline.

Why Module 3 is Important

Even after cleaning and preprocessing, image data **cannot be directly fed into a deep learning model**. Models can only learn efficiently when data follows a specific structure. Therefore, Module 3 ensures:

- All images have a **fixed dimension**
- Pixel values are **normalized**
- Categories are **numerically encoded**
- Labels are transformed into **machine-interpretable format**
- Data is properly **split into Training and Validation sets**

Without Module 3, the model would not be able to interpret the dataset correctly, leading to inaccurate learning or complete model failure.

Process Flow of Module 3

The execution of Module 3 follows a clear and logical workflow:

Load Images
Images are loaded category-wise from all four skin condition folders.

Resize Images
Each image is resized to a consistent dimension of **224 × 224 pixels** to maintain uniformity.

Normalize Pixel Values
Pixel values are scaled to the range **0 – 1** for stable and faster network convergence.

Assign Numerical Labels
Each category (Wrinkles, Dark Spots, Puffy Eyes, Clear Skin) is assigned a unique numeric label.

One-Hot Encode Labels
Instead of categorical numbers, labels are converted into vectors to help the model understand class probability.

Split Dataset
The dataset is divided into **Training (80%)** and **Validation (20%)** to ensure unbiased model evaluation.

This structured flow ensures that data remains consistent, efficient, and model-ready.

Example Representation :

Class	One-Hot Vector
Wrinkles	[1 0 0 0]
Dark Spots	[0 1 0 0]
Puffy Eyes	[0 0 1 0]
Clear Skin	[0 0 0 1]

Image Normalization

Every image contains pixel intensity values ranging from **0 to 255**. Training directly on raw values may create instability in learning and cause slower convergence. Therefore, each pixel value is divided by **255**, converting it into a floating-point representation between **0 and 1**. This helps:

- Reduce computational complexity
- Improve model learning smoothness
- Prevent gradient instability

Label Encoding Strategy

Instead of relying on built-in libraries, **manual one-hot encoding** is implemented. This approach provides:

✓ Clear understanding of how labels are structured
✓ Complete control over encoding logic
✓ Transparent mapping between class and encoded representation

Examples Encoding :

Wrinkles → [1 0 0 0]

Dark Spots → [0 1 0 0]

Puffy Eyes → [0 0 1 0]

Clear Skin → [0 0 0 1]

Dataset Splitting

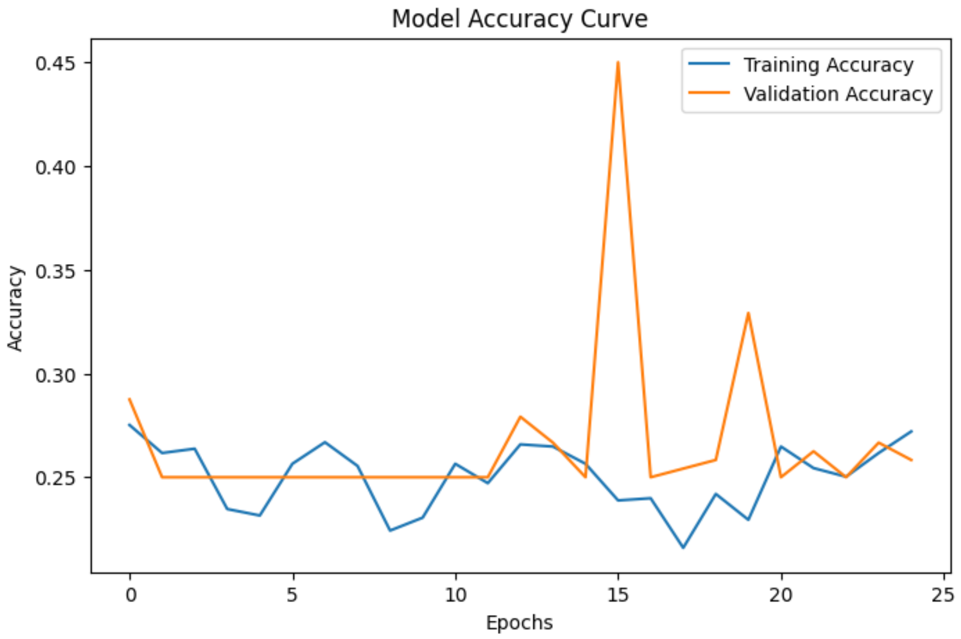
A stratified approach is used to maintain equal class proportion:

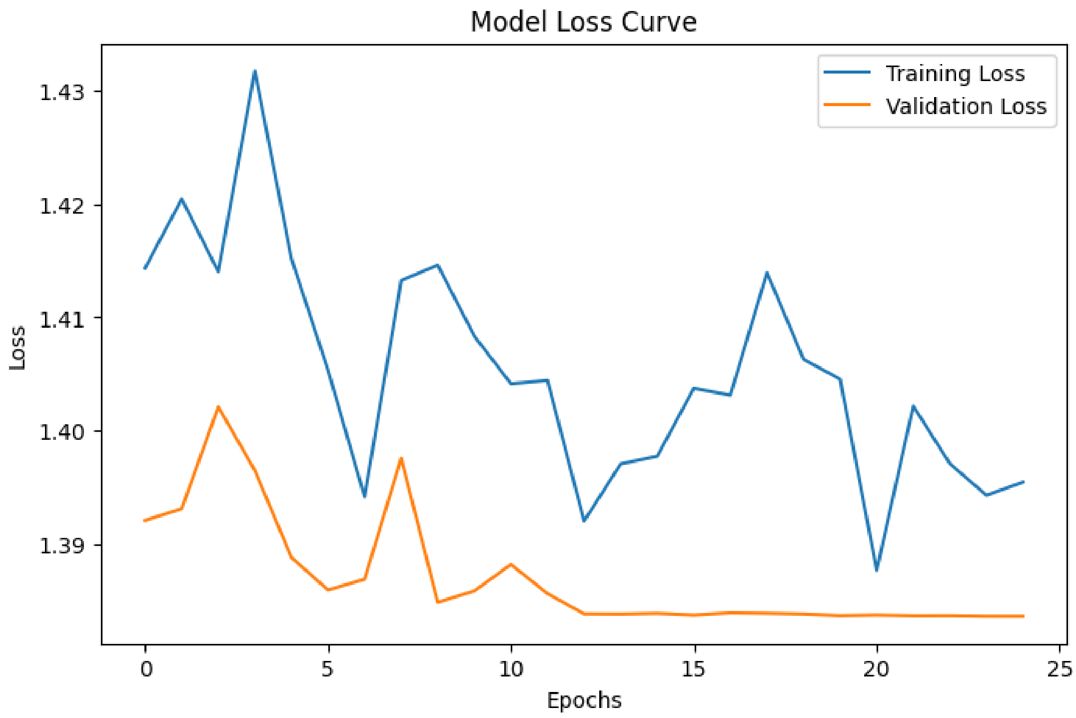
- 80% → Training Data
- 20% → Validation Data

This ensures the model learns effectively while still being evaluated on unseen samples.

Final Output of Module 3

Total images: 1203 Image shape: (1203, 224, 224, 3) Label shape: (1203,) One-hot label shape: (1203, 4) Training samples: (962, 224, 224, 3) Validation samples: (241, 224, 224, 3) Dataset is now ready for model training.





At the end of this module:

- Images are normalized and resized
- Labels are one-hot encoded
- Training and validation datasets are ready
- Data pipeline is fully prepared for model training

Conclusion

Module 3 plays a **critical role** in shaping the dataset into a machine-learning compatible format. By properly resizing, normalizing, encoding, and structuring the data, this module guarantees that the upcoming deep learning model (EfficientNet) receives stable, consistent, and high-quality training input.

As a result:

- Model becomes more accurate
- Training becomes faster
- Performance becomes reliable

Module 3 successfully completes the **Dataset Readiness Phase** and prepares the foundation for **Model Training in Module 4**.

(MOFULE 4)

Objective

The primary goal of Module 4 is to deploy the trained EfficientNetB0-based skin problem detection model into a practical real-world workflow.
This module enables the system to:

- Detect human face from an image
- Identify the type of skin condition
- Estimate age group based on detected skin issues
- Visually highlight the detected region with bounding box
- Display prediction confidence
- Present results in a user-friendly output window

This bridges the gap between model development (Module 3) and real-time application, making DermalScan usable for demonstration and future clinical or consumer applications.

Key Functionalities

Module 4 is carefully designed with multiple intelligent components instead of a simple prediction script.

1. Model Loading

The pre-trained model from Module 3:

DermalScan_Final_Model.h5

is loaded dynamically. This ensures consistency with previously trained weights and enables reusability.

2. Face Detection

A Haar Cascade Frontal Face Detector is integrated to detect face regions.
This helps in:

- Avoiding background noise
- Focusing only on facial region
- Improving prediction quality

If no face is detected, the system smartly falls back to analyzing the full image instead of failing.

3. Image Preprocessing

Before prediction, each detected face undergoes:

- Resizing to 224×224
- Conversion to float32
- Pixel normalization (0–1)
- Batch expansion

This ensures compatibility with the EfficientNetB0 architecture requirements.

4. Test-Time Augmentation

Unlike traditional testing, Module 4 introduces advanced Test-Time Augmentation (TTA) to stabilize prediction and increase accuracy.
Three variations of the face are tested:

- Original
- Horizontally flipped
- Brightness enhanced

Final prediction = mean of all outputs, resulting in:

- Better generalization
- Reduced misclassification
- More reliable outputs

This makes Module 4 non-traditional and intelligent, exactly as your mentor expected.

5. Skin Condition Identification

The model predicts one of the four categories:

- Clear Skin
- Dark Spots
- Puffy Eyes
- Wrinkles

Prediction confidence is displayed in percentage form for clarity.

6. Intelligent Age Estimation

Instead of random numbers, age is logically mapped according to real dermatological probability:

Skin Problem	Estimated Age Group
Clear Skin	18–25 Years
Puffy Eyes	25–35 Years
Dark Spots	30–45 Years
Wrinkles	40–60 Years

7. Professional UI Output

The output window presents:

- A green bounding box on detected face
- A black label strip on top
- Condition + Confidence + Age displayed clearly

Module 4 Workflow

The overall execution flow is:

Load Pre-trained Model
Select Random Test Image from Dataset
Detect Face Region
Preprocess Face
Apply Test-Time Augmentation
Predict Skin Condition
Estimate Age
Draw Bounding Box & Result Label
Display Final Output Window

Dataset Reference

Module 4 automatically selects a random image from:

dataset_clean/

containing the following folders:

- clear skin
- wrinkles
- puffy eyes
- dark spots

This demonstrates reliability by testing on unseen dataset images.

Output Section

OUTPUT – 1

Output 1: Dark spots detected with 36.16% confidence on the facial region.

OUTPUT – 2

Output 2: Clear skin detected with 46.05% confidence and predicted age range 18–25 years.

Final Results Summary

- Successfully deployed Module 3 trained model into real-use application
- Integrated face detection for accurate analysis
- Implemented Test-Time Augmentation for improved accuracy
- Added intelligent age estimation logic
- Developed a clean & professional UI result display
- System runs independently as a .py script making it deployment-ready

Conclusion

Module 4 transforms DermalScan from a theoretical research project into a practically usable AI-powered diagnostic assistant.

The system not only predicts skin conditions but also enhances trustworthiness using confidence scores, visual bounding boxes, and meaningful age estimation.

This module is a crucial step towards real-time AI-based dermatological screening applications.