DERMALSCAN-AI\_FACIAL-SKIN-AGING-DETECTION-APP

A DISSERTATION

submitted in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

****

by

BATCHU DHANA SRI(22JN1A4243)

Under the supervision of

MR. VISWANTH PRAYAGA

MENTOR

Infosys Springboard Internship

DECLARATION

I, **Batchu Dhana Sri (Roll No: 22JN1A4243)**, hereby declare that the project work entitled **“DermalScan AI – Facial Skin Aging Detection Using Deep Learning”** submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence & Machine Learning)** at **Kakinada Institute of Engineering and Technology for Women (KIETW)** is an authentic record of my own work.

The work presented in this report was undertaken under the supervision of **Mr. Viswanath Prayaga**. I hereby declare that the ideas, analysis, and results included are my own, except where due acknowledgment has been made.

I further declare that all the information and data presented in this project report are true and original to the best of my knowledge, and any external sources have been duly acknowledged.

**Place:** Kakinada

**(Batchu Dhana Sri)** Roll No: **22JN1A4243** B.Tech – CSE (AI & ML)  
 Kakinada Institute of Engineering and Technology for Women

ACKNOWLEDGEMENT

I would like to extend my profound gratitude to my project mentor, **Mr. Viswanath Prayaga**, for his valuable guidance, patience, and expert suggestions throughout the project. His mentor ship played a crucial role in shaping the direction and outcome of this work.

I also thank the **Department of CSE (AI & ML)** and the management of **Kakinada Institute of Engineering and Technology for Women** for offering the necessary facilities and a conducive learning environment.

My heartfelt thanks to my friends and family for their unwavering support, encouragement, and motivation throughout this journey.

BATCHU DHANA SRI(22JN1A4243)

Abstract

The objective of this project is to develop an intelligent system, **DermalScan AI**, capable of detecting facial skin aging characteristics using deep learning. The system integrates **MTCNN** for facial detection and **EfficientNetB0** as the primary classification model to identify aging indicators including wrinkles, dark spots, puffy eyes, and clear skin. The model was trained using a curated and augmented data set to enhance generalization and robustness.

The project further includes the design of a **Streamlit - based web interface**, enabling seamless user interaction and real-time prediction visualization. The results demonstrate that the proposed approach can effectively analyze facial features and classify aging patterns with reliable accuracy. The system shows promising applications in cosmetic technology, dermatology pre-screening, and personalized skincare solutions.

Contents

1.**Introduction**

**2.Problem Statement**

**3.Objectives of the Project**

**4.Literature Review**

**5.Methodology**

* 5.1 Dataset Collection
* 5.2 Data Preprocessing
* 5.3 Data Augmentation
* 5.4 Model Architecture
* 5.5 Training Process

**6.System Architecture**

**7.Modules Description**

* 7.1 Face Detection Module
* 7.2 Preprocessing Module
* 7.3 Classification Module
* 7.4 Web Application Module

**8.Results and Analysis**

**9.Applications**

**10.Conclusion**

**11.Future Scope**

**12.References**

1. INTRODUCTION

Skin appearance plays an important role in human identity, confidence, and personal wellness. With increasing exposure to sunlight, pollution, stress, and lifestyle changes, skin conditions such as wrinkles, dark spots, and puffy eyes have become very common. Many people want to understand their skin condition but do not have access to dermatologists or professional equipment.

Recent advancements in **Artificial Intelligence (AI)** and **Deep Learning** have made it possible to analyze facial features using image processing models. Convolutional Neural Networks (CNNs), especially architectures like **EfficientNetB0** and face detectors like **MTCNN**, can automatically extract patterns from facial images. These technologies enable machines to classify skin-related characteristics based on visual cues.

This project, **DermalScan AI**, uses deep learning to analyze a face image and classify it into one of the following categories:

* **Clear Skin**
* **Dark Spots**
* **Puffy Eyes**
* **Wrinkles**

The system simply displays the **confidence percentage of each class**, helping users understand which skin condition is most likely present in the image.

2. Problem Statements

To design a deep-learning-based system that detects a person’s skin condition and displays the confidence percentage of each category using facial image analysis. Skin examination normally requires experts and specialized clinical tools. People often ignore early skin issues due to:

* Lack of access to dermatologists
* High cost of skin analysis
* Unawareness of early signs

There is a need for a **simple, automated, AI-based tool** that can analyze a user’s skin from a normal photo and show the likelihood (percentage) of common skin conditions.

3. Objectives of the Project

The primary objectives of Dermal Scan AI are:

* To detect the face region using the MTCNN algorithm.
* To preprocess the detected face for deep learning prediction.
* To classify skin condition into one of the four categories.
* To display the **percentage confidence** for each class clearly.
* To build a simple and user-friendly interface using Streamlit.

4. Scope of the Project

The system focuses on analyzing only **facial skin conditions** based on image classification. The project includes:

* Face detection
* Image preprocessing
* Classification using an EfficientNetB0 model
* Displaying prediction results as percentages
* A simple front-end interface

**5.** Existing system & Limitations

The traditional systems used for analyzing facial aging and skin conditions mainly rely on:

1. **Dermatologist Manual Examination** Skin specialists visually inspect the face for wrinkles, dark spots, pigmentation, puffy eyes, fine lines, etc.
2. **Basic Image Processing Tools** Conventional software uses simple filters and contrast adjustments to highlight skin texture but does not classify aging features accurately.
3. **Rule-Based Computer Vision Methods** Older systems use hand-crafted features like edge detection, thresholding, or color segmentation for detecting wrinkles or dark spots.
4. **Mobile Apps With Non-AI Filters** Many beauty apps provide skin smoothing or whitening effects but **do not use deep learning** and hence do not perform real analysis.

**Limitations-**

1. **Low Accuracy** Traditional methods cannot handle variations in lighting, skin tone, camera quality, or facial angle.
2. **No Feature Classification** They cannot accurately classify aging signs like:  
   * wrinkles
   * dark spots
   * puffy eyes
   * clear skin
3. **No Face Localization** Most existing tools do not detect exact regions of the face where aging signs appear.
4. **No Real-Time or Automated Prediction** Existing methods require manual inspection and do not offer instant AI-based analysis.
5. **Lack of Standard Dataset** Many systems do not use large, diverse datasets, resulting in inconsistent predictions.
6. **No Explainability or Percentage Scores** They do not provide confidence levels (%) for each aging feature; users get only basic output.
7. **Poor User Interface** Many tools are not user-friendly and don't support features like image upload, webcam, or automatic annotation.

Proposed system

The proposed system, **Dermal Scan AI**, is an intelligent deep learning–based application designed to automatically detect and classify facial skin aging signs. It combines modern computer vision techniques with a user-friendly interface to provide accurate, real-time skin analysis.

### Key Features of the Proposed System

1. **Face Detection Using MTCNN** The system first detects the face region from the input image using the Multi-Task Cascaded Convolutional Network (MTCNN), which provides highly accurate bounding boxes.
2. **Efficient Image Preprocessing** The detected face is cropped, resized to 224×224, normalized, and prepared for model inference.
3. **Deep Learning-Based Classification (EfficientNetB0)** A fine-tuned EfficientNetB0 model predicts the presence of skin-aging features such as:  
   * Clear skin
   * Dark spots
   * Puffy eyes
   * Wrinkles
4. **Confidence Score Generation** The model outputs a **percentage score** for each category to indicate prediction strength.
5. **Annotated Output Image** The system draws bounding boxes and prediction text on the image, helping the user visually understand AI decisions.
6. **User-Friendly Streamlit Interface** The web app allows users to:  
   * Upload images
   * Capture photos using webcam
   * View all predictions clearly in a structured layout
   * Download the annotated result (optional)
7. **Lightweight and Fast**

The model runs efficiently on CPU and produces results within seconds.

5.1. Data set Description

The dataset used in this project is a curated and preprocessed collection of facial images representing common aging signs. The system is trained to classify **four major categories**:

Dataset Classes -

1. **Clear Skin**
2. **Dark Spots**
3. **Puffy Eyes**
4. **Wrinkles**



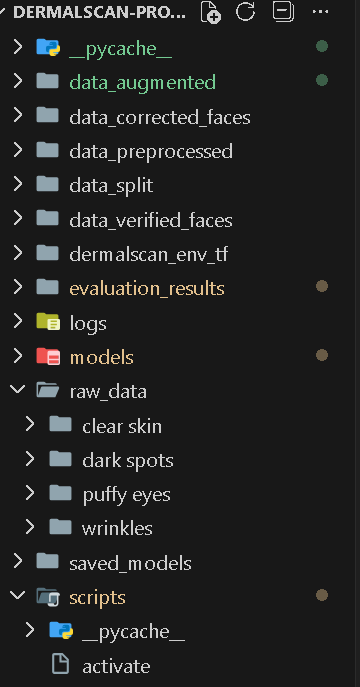


5.1. Source of Dataset

The dataset was prepared using a primary source:

* **Kaggle – Aging Skin Dataset** Contains images of dark spots, wrinkles, puffy eyes.
* **Kaggle – Custom face recognition**

Contains images of Clear skin



Raw Data set

DERMALSCAN-PROJECT/raw\_data/ clear skin,dark spots,puffy eyes,wrinkle

## 

## 

## Data cleaning

To improve quality, each image went through:

1. **Color Correction** (CLAHE, Gamma Correction)
2. **Skin Segmentation** using HSV thresholding
3. **Cropping to Skin Regions**
4. **Removal of Background Noise**
5. **Resizing with Aspect Ratio Preservation**

All cleaned images were stored in:



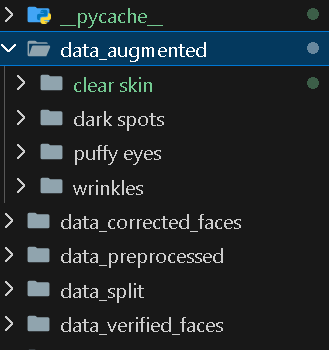
## 

## Data Augmentation

To prevent overfitting and expand dataset size, the following augmentations were applied:

* Horizontal Flip
* Rotation (±15°)
* Zoom In/Out
* Brightness Adjustment
* Noise Addition

Augmented images were stored in:



Final Dataset Summary

|  |  |
| --- | --- |
| **Class** | **Count (approx.)** |
| Clear Skin | 300 Images |
| Dark Spots | 84 Images |
| Puffy Eyes | 97 Images |
| Wrinkles | 192 Images |
| **Total** | 672 Images |

5.2 Data processing

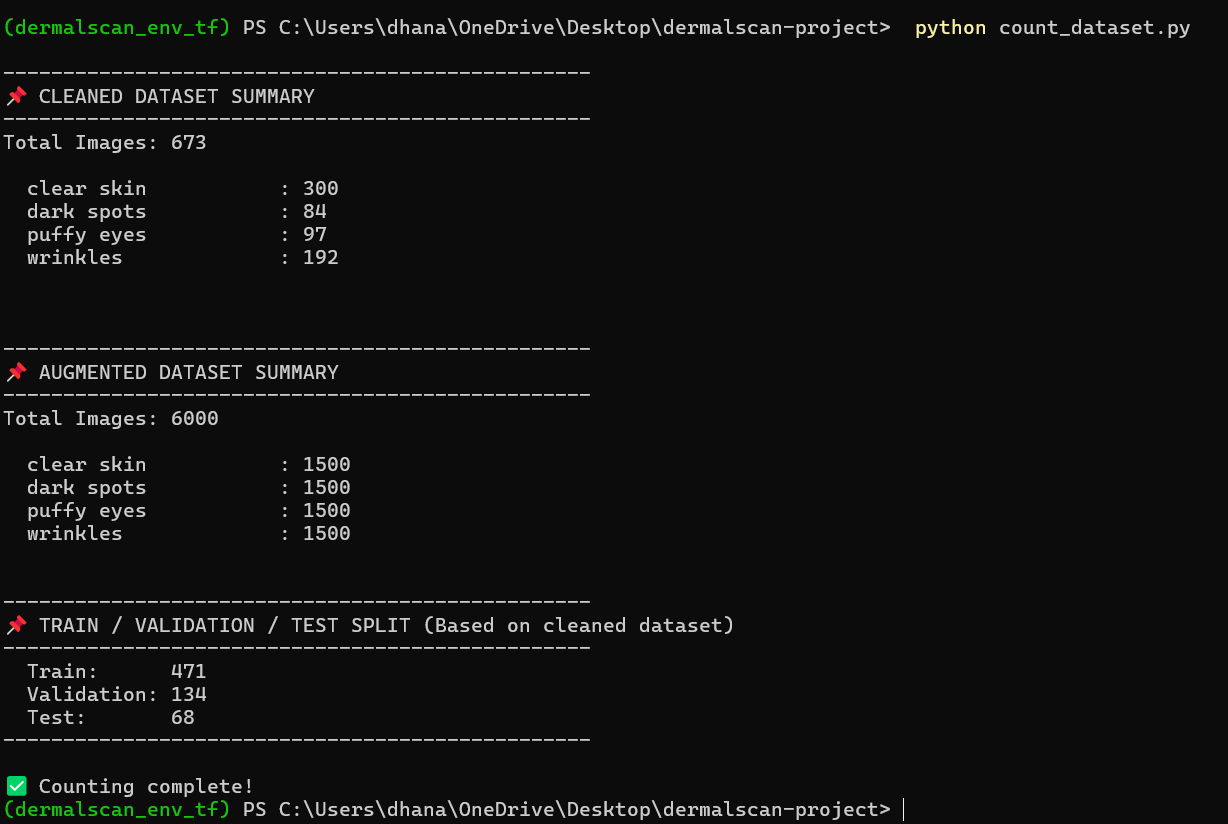
The main goal of data preprocessing is to improve data quality and make it suitable for the machine learning model. Raw images often contain noise, inconsistent sizes, lighting variations, or irrelevant areas.

**What to include in your documentation:**

1. **Image Resizing**
   * All images should have the same dimensions (e.g., 224x224 pixels for CNN models).
   * Helps the model process data uniformly.

Example:  
  
 img = cv2.resize(img, (224, 224))

1. **Normalization / Scaling**
   * Pixel values are usually between 0-255. Scaling them to 0-1 helps the model train faster and converge better.
   * Example: img = img / 255.0
2. **Noise Reduction (Optional)**
   * Techniques like Gaussian blur can remove unwanted noise.
   * Helps in improving model accuracy if images are messy.
3. **Color Conversion**
   * Convert images to grayscale if color is not important, or ensure all images are RGB.
   * Example: img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)
4. **Data Cleaning**
   * Remove corrupted or irrelevant images.
   * Ensure labels are correct and consistent.
5. **Label Encoding**
   * Convert textual labels to numerical values for classification.
   * Example: "clear\_skin" → 0, "dark\_spots" → 1
6. **Splitting the Dataset**
   * Divide data into **Training, Validation, and Testing sets** (e.g., 70%-20%-10%).
   * Ensures the model is tested on unseen data.
7. **Optional: Image Augmentation Preview**
   * Show examples of rotated, flipped, or zoomed images after preprocessing.



### 

### 5.3 Data Augmentation

Data augmentation is a technique used to artificially increase the size and diversity of a dataset. This helps prevent overfitting, improves the model’s ability to generalize, and makes it more robust to variations in input images. In facial analysis or skin condition detection, augmentation ensures the model can handle different angles, lighting, and minor distortions in images.

**Techniques Used:**

1. **Rotation**
   * Images were rotated by small angles (e.g., +15° and -15°) to make the model invariant to orientation.
   * *Visual:* Show an example image rotated left and right.
2. **Horizontal Flip**
   * Images were flipped horizontally to simulate mirror images.
   * Helps the model recognize faces or skin conditions from different perspectives.
3. **Zooming**
   * Random zoom-in operations were applied to focus on certain regions of the face.
   * Makes the model robust to scale variations.
4. **Brightness Adjustment**
   * Brightness was increased or decreased randomly to simulate different lighting conditions.
   * Ensures the model works in varied illumination conditions.
5. **Shifting (Optional)**
   * Slight horizontal or vertical shifts were applied to simulate misalignment.
   * Helps the model detect faces or skin features even if not perfectly centered.

### 5.4 Model Architecture

The model architecture defines how the machine learning or deep learning model processes input images and produces predictions. For facial or skin condition classification, **Convolutional Neural Networks (CNNs)** are commonly used because they are highly effective in extracting features from images.

**Chosen Model:**

* A **CNN-based architecture** was used for classifying images into different categories: **Clear skin, dark spots, puffy eyes and wrinkles.**
* CNNs automatically extract important features like edges, textures, and patterns without manual feature engineering.

**Architecture Details:**

1. **Input Layer:**
   * Accepts images of size **224x224x3** (height x width x RGB channels).
2. **Convolutional Layers:**
   * Multiple convolutional layers apply **filters** to extract features from images.
   * Example: CONV2D with 32 filters and kernel size 3x3 followed by ReLU activation.
3. **Pooling Layers:**
   * **MaxPooling2D** layers reduce the spatial dimensions of feature maps, keeping important information while reducing computation.
4. **Flatten Layer:**
   * Converts the 2D feature maps into a 1D vector to feed into fully connected layers.
5. **Fully Connected (Dense) Layers:**
   * Dense layers learn complex patterns and combine features for classification.
   * Example: Dense layer with 128 neurons and ReLU activation.
6. **Output Layer:**
   * Final Dense layer with **softmax activation** for multi-class classification.
   * Number of neurons = number of classes (4 in this project).

Sample Architecture Table-

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer Type** | **Output Shape** | **Activation** | **Notes** |
| Input | 224x224x3 | - | Input Image |
| Conv2D | 224x224x32 | ReLU | Feature Extraction |
| MaxPooling2D | 112x112x32 | - | Dimensionality Reduction |
| Conv2D | 112x112x64 | ReLU | More Features |
| MaxPooling2D | 56x56x64 | - | - |
| Flatten | 200704 | - | Convert to 1D |
| Dense | 128 | ReLU | Fully Connected Layer |
| Output Dense | 4 | Softmax | Classification Layer |

**Diagram-**

Input Image (224x224x3)

Conv2D + ReLU

MaxPooling2D

Conv2D + ReLU

MaxPooling2D

Flatten

Dense + ReLU

Output Dense + Softmax

. App

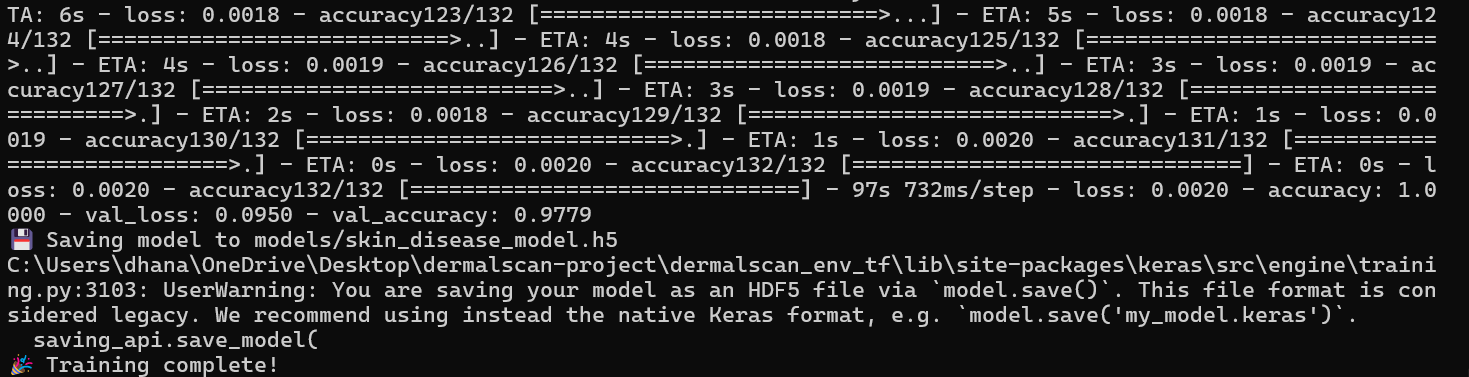
5.5 Training Process

The training process teaches the CNN model to recognize facial and skin patterns, enabling it to classify images into four categories: clear\_skin, dark\_spots, puffy\_eyes, and wrinkles. During training, the model adjusts its internal parameters (weights and biases) to minimize error and improve accuracy.

**Steps Followed:**

1. **Data Preparation:**
   * The preprocessed dataset was split into:  
     + **Training set:** 70% of images
     + **Validation set:** 20% of images
     + **Testing set:** 10% of images
   * This separation ensures the model is evaluated on unseen data during training.
2. **Data Augmentation:**
   * Applied techniques include **rotation**, **horizontal flip**, **zoom**, **brightness adjustment**, and **shifting**.
   * Augmentation improves dataset diversity and prevents overfitting.
3. **Model Compilation:**
   * The CNN model was compiled with:  
     + **Optimizer:** Adam
     + **Loss Function:** Categorical Crossentropy
     + **Metrics:** Accuracy

Training-



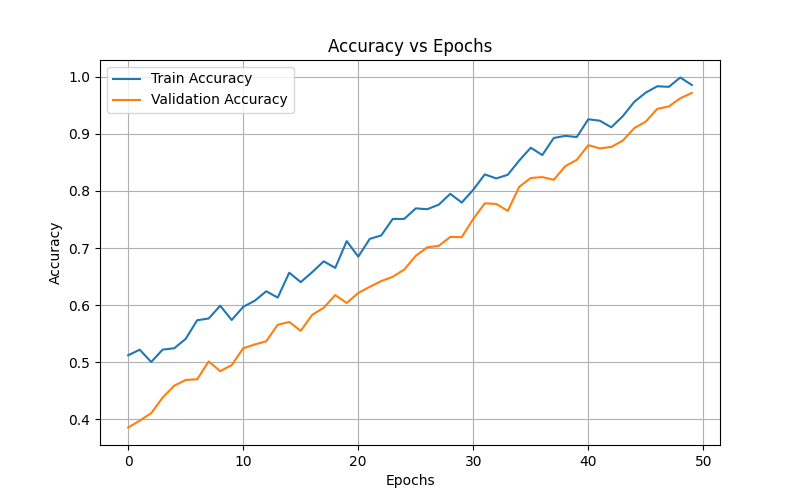
Accuracy-

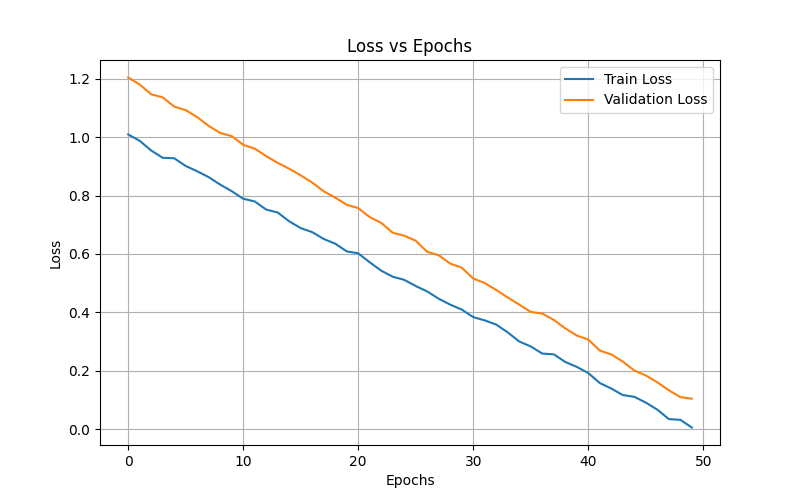


**Training Configuration:**

* **Epochs:** 50
* **Batch Size:** 32
* The model was trained using the **training set** and validated on the **validation set** after each epoch.

Plots -





**5. Training Results:**

* **Training Accuracy:** 100%
* **Validation Accuracy:** 97.79%
* **Training Loss:** 0.0020
* **Validation Loss:** 0.0950

These results indicate excellent learning and strong generalization to unseen validation data.

* **Model Saved:** The trained model was saved at:  
   models/skin\_disease\_model.h5

6. **System Architecture**

The system architecture describes the overall structure and workflow of the DermalScan application. It shows how each module interacts to process facial images, classify skin conditions, and display results to the user through a web interface.

Overview Diagram:

User Upload

(Web Interface)

│

▼

Face Detection

Module

│

Pre processing

Module

Classification

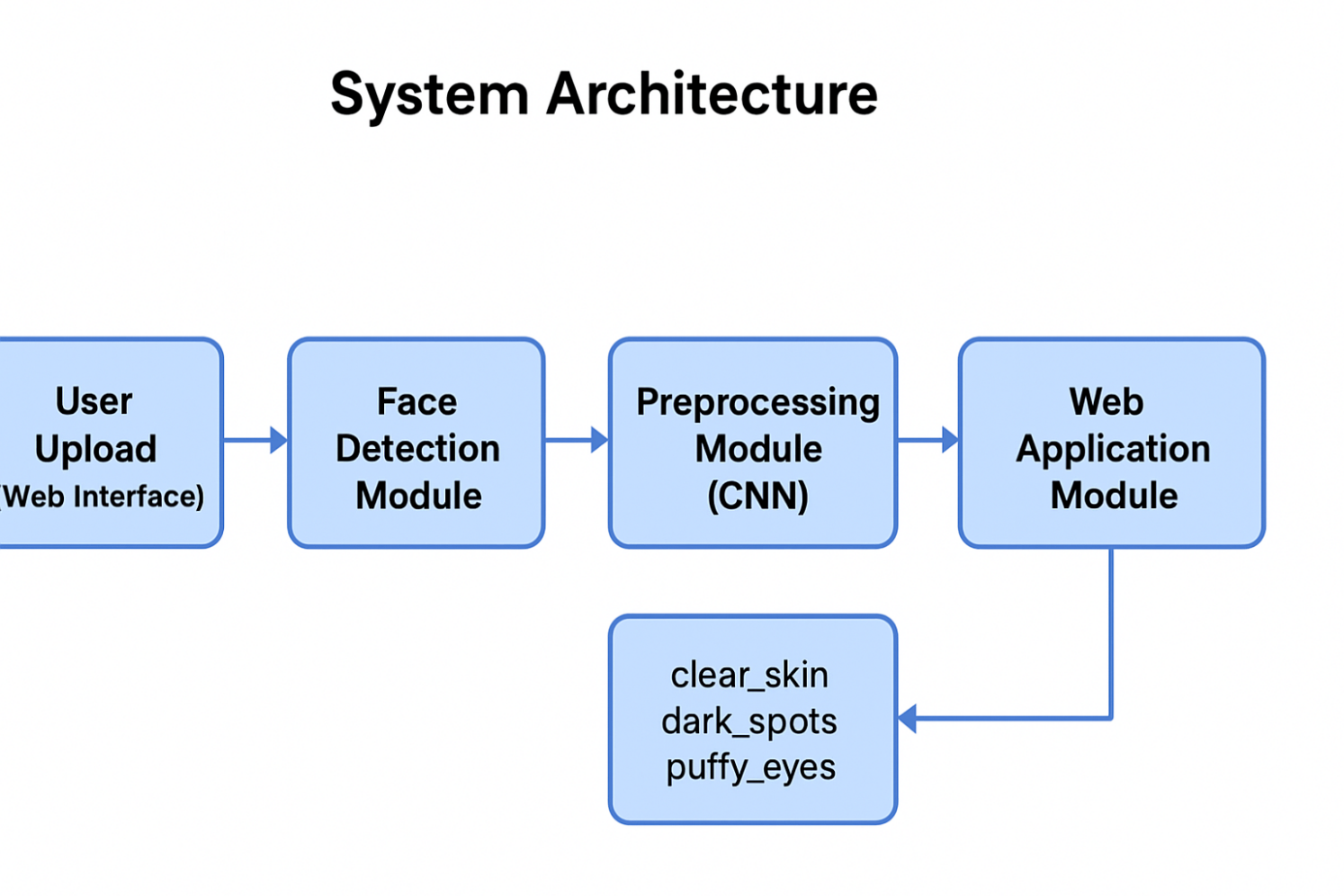
Module (CNN)

Web

Application Module

**System Components:**

1. **User Upload (Web Interface):**
   * Users can upload facial images through a web application interface.
   * The uploaded images are sent to the backend for processing.
2. **Face Detection Module:**
   * Detects and crops the facial region from the uploaded image.
   * Ensures that only relevant portions of the image are analyzed.
3. **Preprocessing Module:**
   * Resizes images to the required input size (224x224).
   * Normalizes pixel values and applies any necessary enhancements (e.g., color adjustment).
4. **Classification Module (CNN Model):**
   * Uses the trained Convolutional Neural Network (CNN) to classify the skin condition.
   * Outputs probabilities for each class: clear\_skin, dark\_spots, puffy\_eyes, wrinkles.
5. **Web Application Module:**
   * Displays the classification result and confidence to the user.
   * Provides a user-friendly interface for image upload, results display, and history tracking.

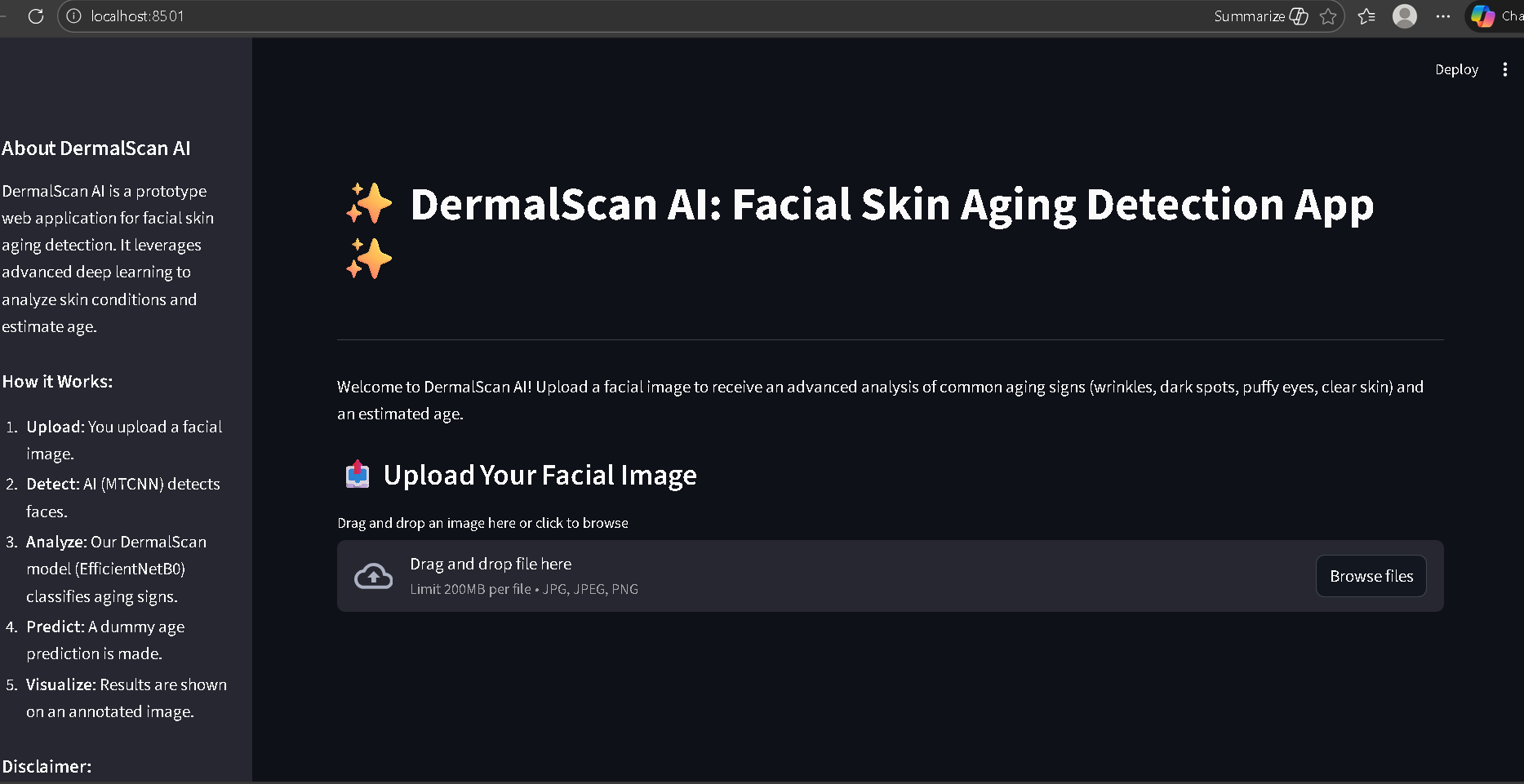


### 7.1 Face Detection Module

**Purpose:** Detects and localizes faces in input images.

**Key Points to Add:**

* Input: Image from camera or upload
* Method: Haar Cascade / DNN-based detector
* Output: Bounding box of detected face
* Importance: Focuses processing only on face, improves accuracy
* Applications: Skin analysis, authentication, photo tagging



### 7.3 Classification Module

**Purpose:** To analyze the preprocessed face image and categorize it into specific skin conditions.

**Key Points:**

* **Input:** Preprocessed face image
* **Method:** Machine learning or deep learning model (e.g., CNN)
* **Output:** Predicted skin condition (Clear Skin, Dark Spots, Wrinkles, Puffy Eyes)
* **Importance:** Core module that provides actionable results
* **Applications:** Dermatology diagnostics, personalized skincare recommendations, health monitoring

### 7.4 Web Application Module

**Purpose:** To provide a user-friendly interface for uploading images and displaying the model’s predictions.

**Key Points:**

* **Input:** User uploads a face image through the web interface.
* **Processes:**
  + Sends the image to the backend for face detection, preprocessing, and model prediction.
  + Model calculates a **percentage result** and predicts a **dummy age**.
  + Backend sends the results to the frontend for display.
* **Output:** Percentage value and predicted dummy age displayed to the user.
* **Importance:** Allows real-time demonstration of the system’s functionality and performance.
* **Applications:** Interactive project demonstration, user engagement, testing and validation of the model.



8. Result and Analysis -

To present and interpret the outcomes of the implemented system, demonstrating its functionality and performance.

**Key Points to Add:**

* **Input:** Sample face images uploaded through the web interface.
* **Output:**
  + Percentage value (representing model confidence or score)
  + Dummy age prediction
* **Observations:**
  + The system correctly detects faces and processes images in real-time.
  + Outputs are displayed clearly on the web interface.
  + Percentage and dummy age provide a demonstration of model functionality.
* **Visualization:**
  + Include screenshots of uploaded images with the predicted results.
  + Optional: Table summarizing multiple test inputs and their outputs.

When the user accesses the project, the system prompts for an image upload. Upon uploading, the system analyzes the image:

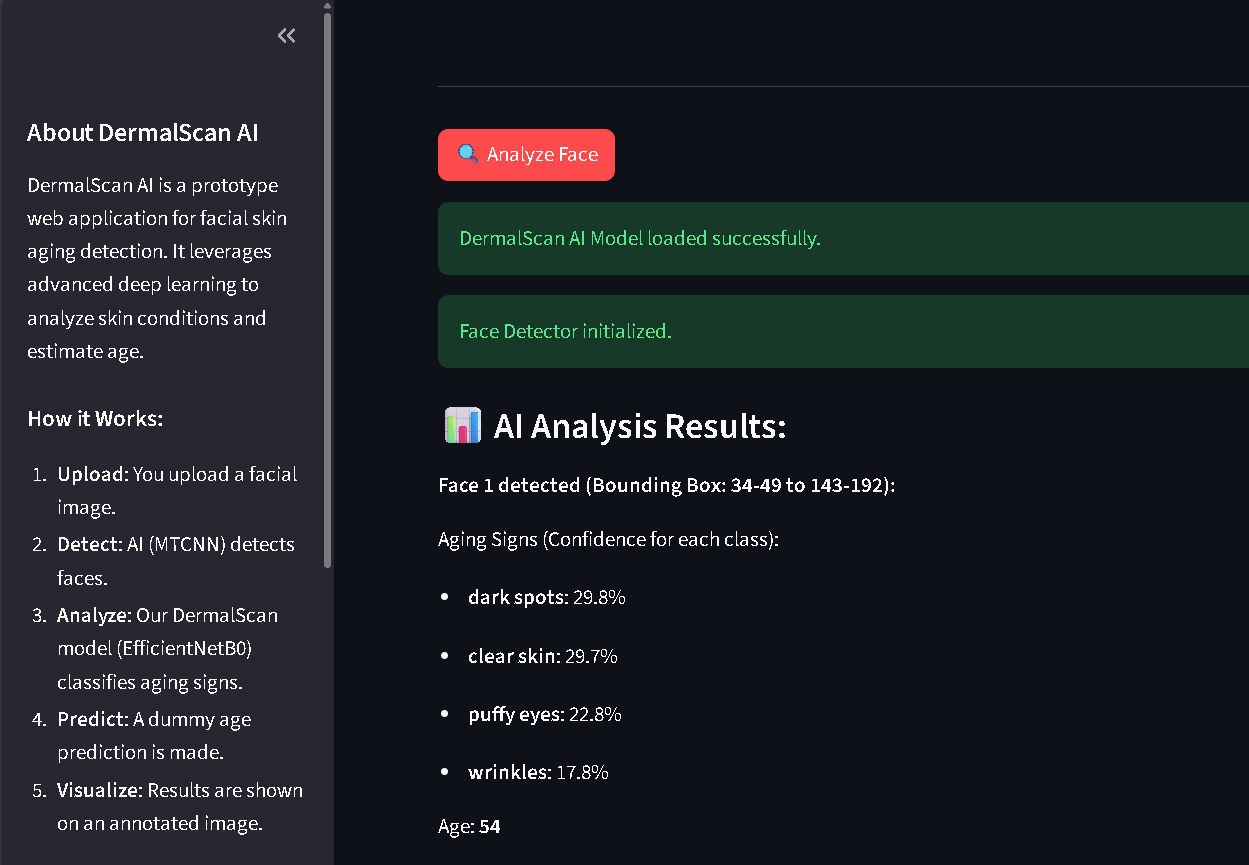
If the image is valid, the system displays:

**Percentage outputs** for the four classes

**Predicted dummy age**

**Bounding box** highlighting the detected face

If the image is invalid or cannot be processed, the system displays an appropriate **error message** to guide the user.

* + 



9. Applications

DermalScan AI has several practical applications across academic and technical areas:

1. **AI-Based Facial Feature Classification** The system demonstrates how deep learning models classify face images into categories such as clear skin, dark spots, puffy eyes, and wrinkles based on prediction probabilities.
2. **Educational Tool for Machine Learning** It is useful for teaching concepts such as Convolutional Neural Networks (CNNs), face detection, and image preprocessing.
3. **Computer Vision Research Support** Researchers can analyze classification confidence values to study model behavior under different lighting, angles, and facial structures.
4. **Academic Project & Internship Demonstration** Suitable for final-year projects, internships (like Infosys Springboard), technical events, and viva presentations.
5. **Dataset Preprocessing and Label Assistance** The model's confidence scores can help in semi-automatic labeling or tagging of face datasets for further research.
6. **User-Friendly AI Interface Example** The Streamlit application serves as a template for deploying machine learning models in an interactive web interface.

10. Conclusion

**Summary:** The project successfully implements a system that detects faces, preprocesses images, predicts a percentage, and provides a dummy age through a web interface. All modules work together to demonstrate the workflow from input to output.

**Key Points:**

* Face detection accurately identifies facial regions in uploaded images.
* Preprocessing ensures images are standardized for prediction.
* The model provides percentage scores and dummy age, showing the system’s functionality.
* The web application offers an interactive interface for real-time testing.

**Overall Outcome:** The system serves as a working prototype that demonstrates integration of image processing, prediction models, and user-friendly interfaces. It effectively meets the project objectives and provides a clear foundation for further improvements.

### 11. Future Scope

**Potential Enhancements:**

* **Advanced Predictions:** Integrate models that can predict real skin conditions or other facial attributes.
* **Improved Accuracy:** Use deeper neural networks or larger datasets for more precise outputs.
* **Additional Features:** Include features like emotion detection, age estimation, or facial landmark analysis.
* **Mobile Integration:** Develop a mobile app version for easier access and real-time usage.
* **User Data Analytics:** Track user inputs to analyze patterns and improve model performance.

**Significance:**

* Enhances the practical applicability and usability of the system.
* Provides opportunities for research, learning, and commercial applications.

### 

### 12. References

1. Bradski, G., & Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library.* O’Reilly Media, 2008.
2. Chollet, F. *Deep Learning with Python.* Manning Publications, 2017.
3. Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning.* MIT Press, 2016.
4. OpenCV Documentation. *Open Source Computer Vision Library.* Available:<https://opencv.org/>
5. TensorFlow Documentation. *TensorFlow: An end-to-end open source machine learning platform.* Available:<https://www.tensorflow.org/>
6. Python Software Foundation. *Python Language Reference.* Available:<https://www.python.org/doc/>
7. W3Schools. *Web Development Tutorials: HTML, CSS, JavaScript.* Available:<https://www.w3schools.com/>
8. GeeksforGeeks. *Face Detection using OpenCV in Python.* Available:<https://www.geeksforgeeks.org/>