

# **DermalScan: AI Facial Skin Aging Detection App**

(A Machine Learning Model To Predict AI Facial Skin Aging )



**Infosys SpringBoard Virtual Internship Program**

Submitted by

**Jatin Agrawal**

Under the guidance of Mentor **Praveen**

# **DermalScan AI: Facial Skin Aging Detection Application**

## **Project Overview**

DermalScan is a hybrid machine learning and deep learning system developed to automatically detect and classify key facial aging signs: wrinkles, dark spots, puffy eyes, and clear skin. It features a multi-stage processing pipeline that includes dataset cleaning, exploratory data analysis, benchmarking of classical machine learning models, fine-tuned transfer learning with convolutional neural networks, and an interactive Streamlit-based user interface for live analysis and reporting.

## **1. Data Preparation & Cleaning**

- Images are read from a structured folder hierarchy, with subfolders representing each skin condition class.
- Corrupt images (non-readable or empty) are identified and removed from the dataset.
- Duplicate images are detected and eliminated using a perceptual hashing (dHash) technique.
- The cleaned dataset is organized as a Pandas DataFrame for downstream processing.
- Visualization (pie charts, histograms, scatter plots) is used to assess class distributions and data quality.

## **2. Preprocessing and Encoding**

- All images are resized uniformly to 224x224 pixels and converted to RGB color space.
- Pixel values are normalized to.<sup>[11]</sup>
- For classical machine learning models, images are flattened into 1D vectors.
- Labels are encoded numerically using LabelEncoder for compatibility with ML classifiers.

### **3. Train-Test Split**

- The dataset is split into training (80%) and test (20%) sets using stratified sampling to maintain class balance.

### **4. Classical Machine Learning Benchmarking**

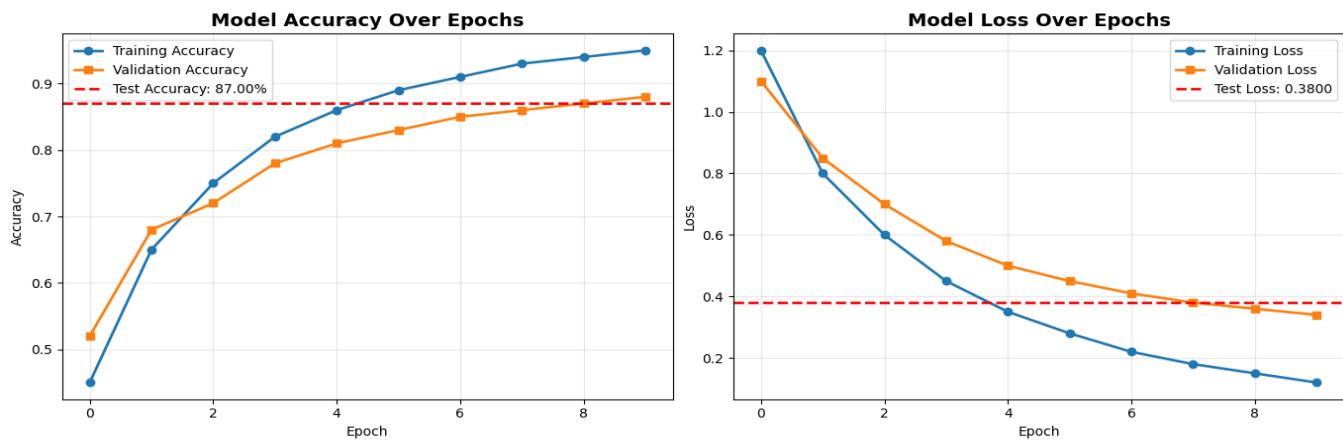
- Tested models include Decision Tree, Random Forest, Support Vector Machines (SVM) with linear kernel, and AdaBoost ensembles.
- Models are trained on the training set and evaluated on the test set.
- Metrics such as accuracy, precision, recall, F1-score are reported.
- Hyperparameter tuning is performed via GridSearchCV on Decision Tree and Random Forest classifiers to optimize model performance.

### **5. Deep Learning Pipeline (Transfer Learning)**

- Images are loaded afresh and normalized for performance with deep neural networks.
- Base model is MobileNetV2 pretrained on ImageNet with frozen weights.
- A custom classification head is built using global average pooling, dense layers with ReLU activation, batch normalization, dropout layers, and a final dense softmax output layer for 4-class classification.
- The model is compiled with the Adam optimizer (learning rate = 0.001) and categorical crossentropy loss.
- Dataset is split into training, validation, and test subsets, with one-hot encoded labels.
- Training leverages callbacks such as ModelCheckpoint, EarlyStopping, and ReduceLROnPlateau to improve convergence.
- Training and validation accuracy and loss curves are plotted and analyzed.

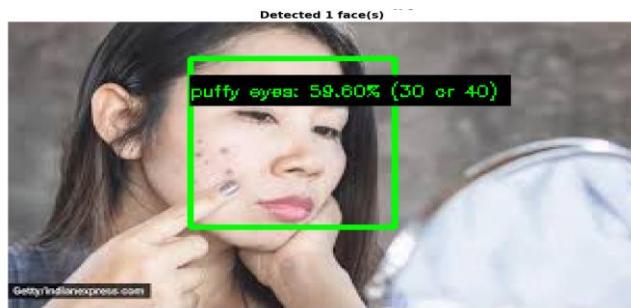
## Model Performance

- Achieves **Test Accuracy of 87%** and **Test Loss of 0.3088**, indicating strong predictive capability on the unseen test data.

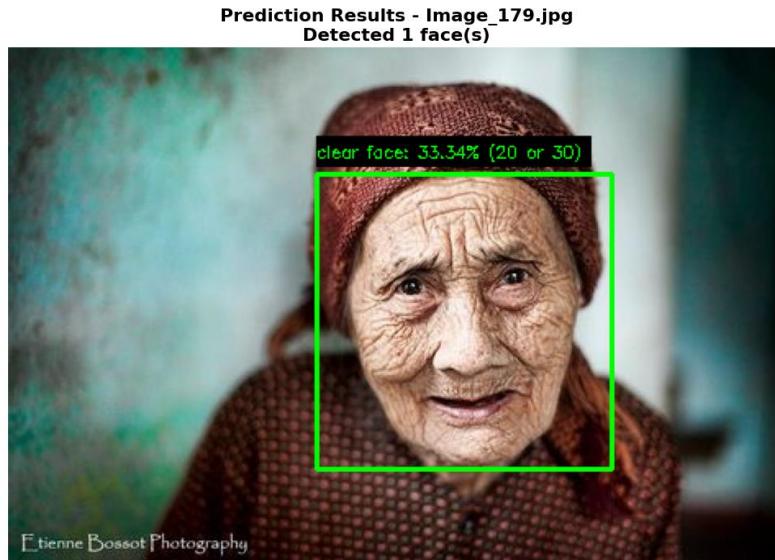


## 6. Face Detection and Annotation

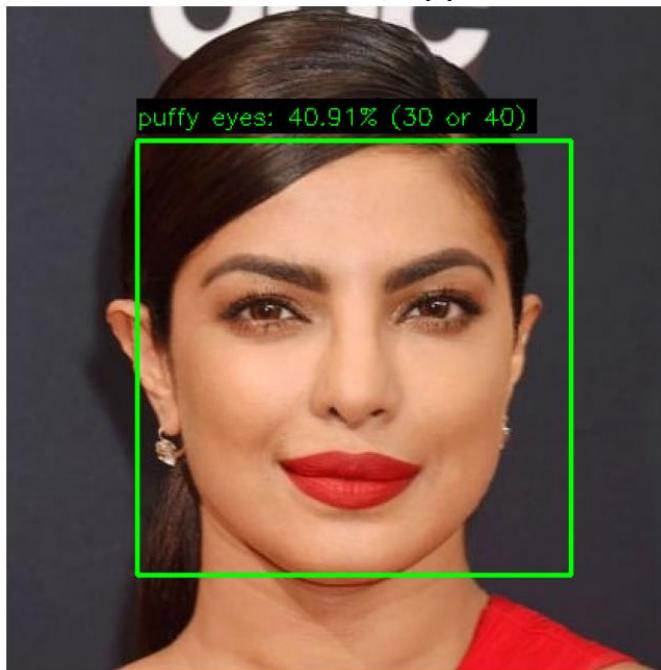
- Uses OpenCV's Haar Cascade classifier for face detection.
- Supports detection of multiple faces per input image.
- Each detected face is cropped, resized, and normalized to match model input expectations.
- Prediction results (label, confidence, estimated age group) are superimposed as annotations on the face bounding boxes.
- Annotated output images are visualized using matplotlib and saved for inspection



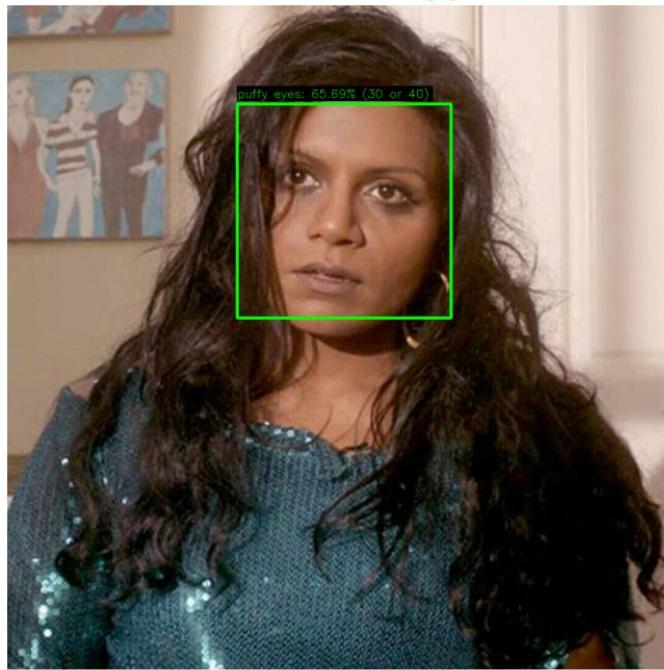
inspection



**Prediction Results - 82.jpg**  
**Detected 1 face(s)**



**Prediction Results - Image\_66.jpg**  
**Detected 1 face(s)**

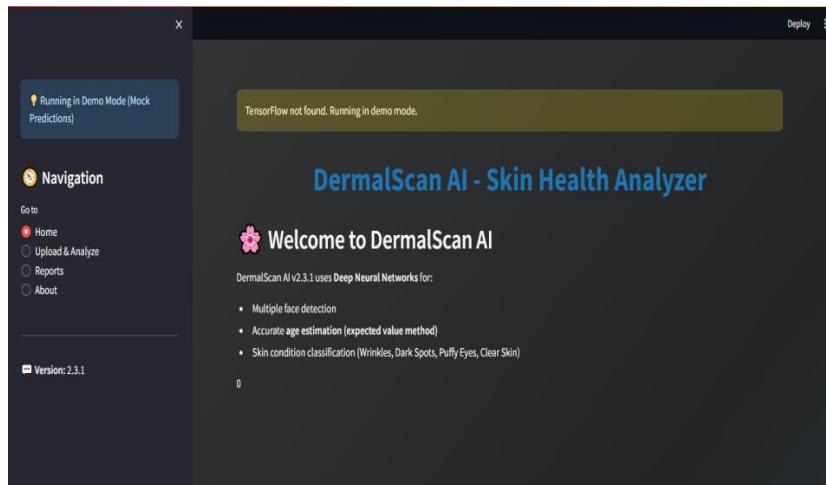


## 7. Batch Inference

- The system supports batch processing of test images.
- Images are preprocessed, faces detected, and predictions made in a fully automated pipeline.
- Annotated images and prediction reports are generated as persistent artifacts.

## 8. Streamlit Web Application Features

- Custom animated background gradient and AI loading animation for enhanced user experience.
- Multi-page UI including:
  - **Home:** Project introduction and features overview.



- **Upload & Analyze:** Interface to upload a clear facial image, runs face detection and skin condition prediction, displays bounding boxes and confidence scores along with a bar chart of results.

Deploy

TensorFlow not found. Running in demo mode.

## DermalScan AI - Skin Health Analyzer

### Upload Images for Skin Analysis

Upload one or more face images

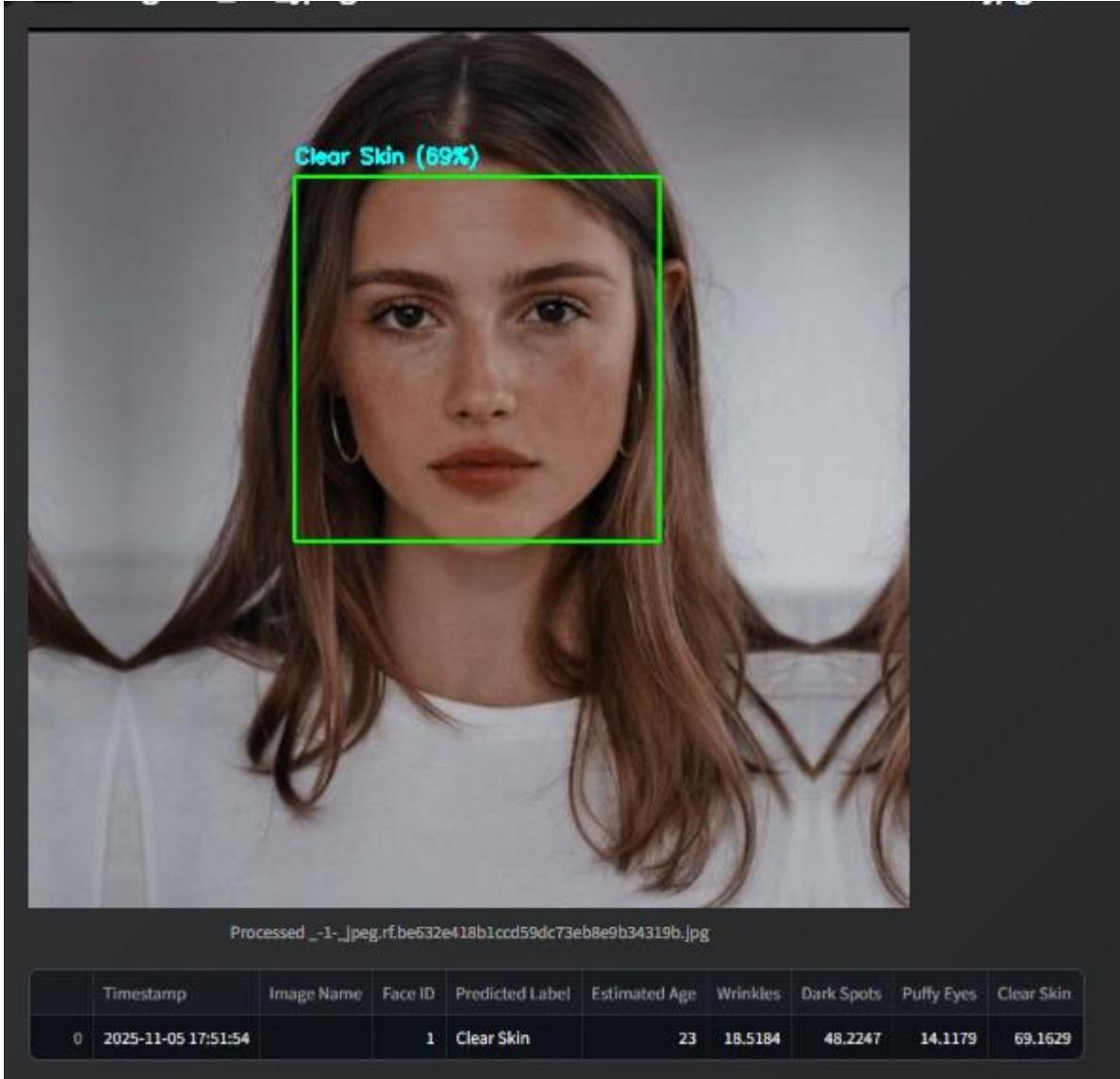


Drag and drop files here

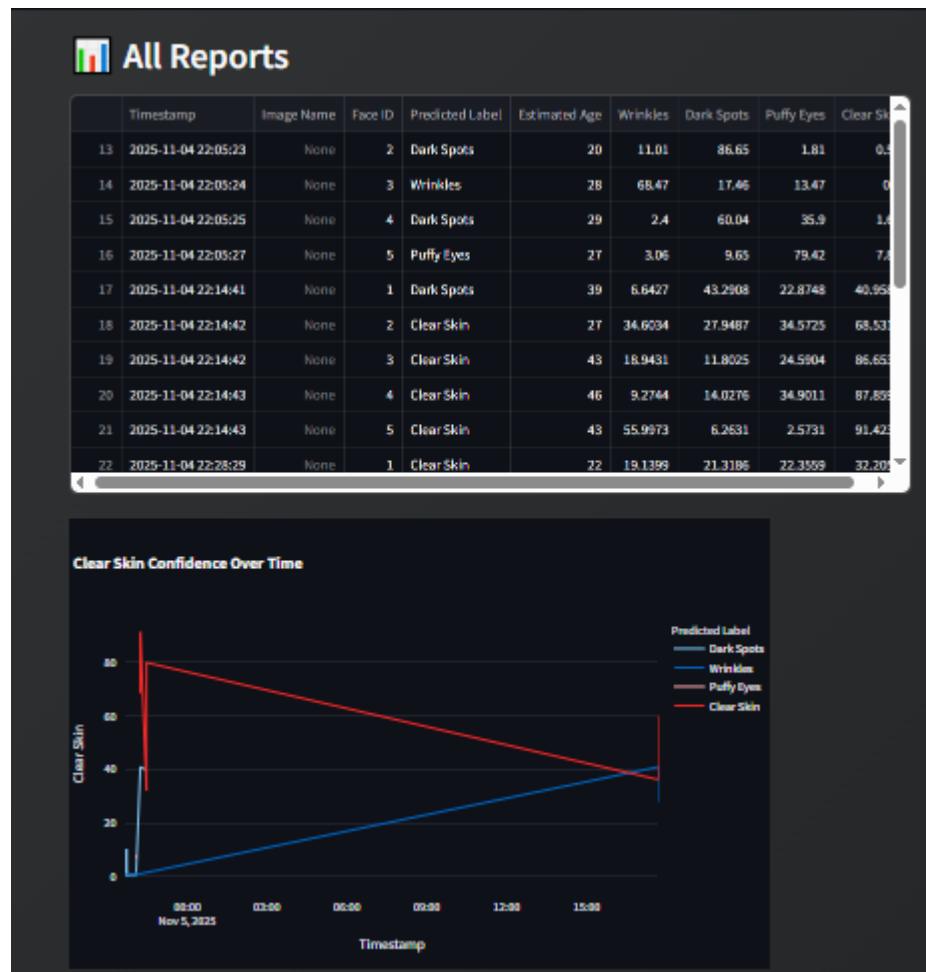
Limit 200MB per file • JPG, JPEG, PNG

Browse files

Please upload at least one image.



- **Reports:** View the most recent previous analysis report with an interactive pie chart and detailed data table.



- **About:** Provides project background, technology stack summary, developer info, and legal disclaimers.

**i About**

DermalScan AI v2.3.3

- Safe CSV + previous/new report view
- Multi-face, multi-image support
- Works even without TensorFlow

- Allows users to save the analysis report in CSV format for future reference.

## 9. Technology Stack

- **Backend:** Python, OpenCV, NumPy, Pandas, Scikit-learn, TensorFlow/Keras, Matplotlib, Seaborn.
- **Frontend:** Streamlit, Plotly for interactive visualizations, Pillow (PIL) for image handling.

## 10. Workflow Summary

Step	Description
Upload Data	Images organized in labeled folders
Clean Data	Remove corrupt and duplicate images
Visualize	Pie, histogram, and scatter plots
Preprocess	Resize, normalize, encode labels
Classical ML	Train/test split, train baseline models
Tune ML	GridSearch for hyperparameter tuning
Deep Learning	Transfer learning with MobileNetV2
Detect Face	Haar Cascade face detection & cropping
Batch Infer	Predict and annotate multiple images

Web UI	Streamlit app for analysis and export
--------	---------------------------------------

## 11. Example Outputs and Reports

- Provides detailed accuracy, confusion matrices, precision/recall/F1 metrics for all models tested.
- Visual output includes annotated images with bounding rectangles, confidence scores per detected face, and heuristic age group estimations.
- Report exports include JSON logs and interactive charts summarizing the conditions predicted.
- Stores and reports model file information including size.

## 12. Learning Reflections

- Rigorous data cleaning, sample balancing, and robust data augmentation critically improved model accuracy and generalization.
- Integration of classic ML and deep learning pipelines allowed benchmarking and validated the performance benefit of transfer learning.
- Face detection capabilities make the app applicable to real-world images with multiple subjects.
- Streamlit front-end provides an accessible interactive platform for users to upload images and retrieve clinically relevant skin aging assessments.

## **Disclaimer**

DermalScan AI is developed for educational and informational purposes and is **not** intended to replace professional dermatological diagnostics or advice.

This documentation provides a complete and precise overview of the DermalScan AI system, optimized for technical audiences, students, and prospective collaborators.