DermalScan: AI Facial Skin Aging Detection App

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

Infosys SpringBoard Virtual Intership Program

Submitted by

Jatin Agrawal

Under the guidance of Mentor **Praveen**

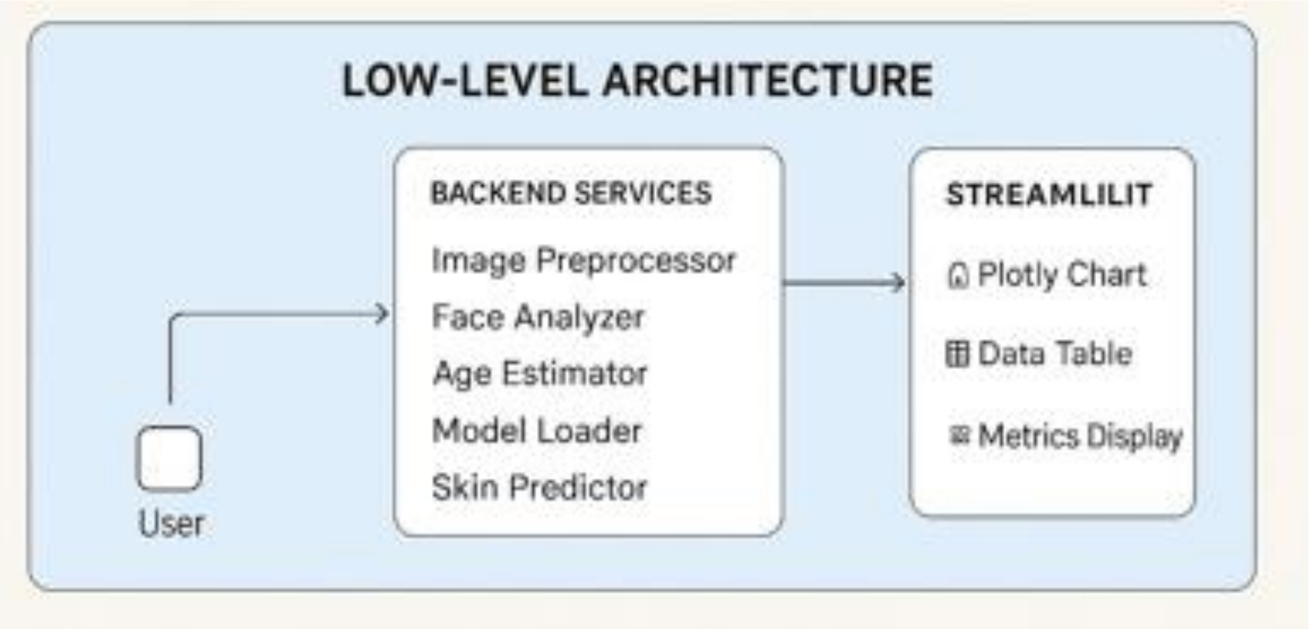
DermalScan AI: Facial Skin Aging Detection Application

Project Overview

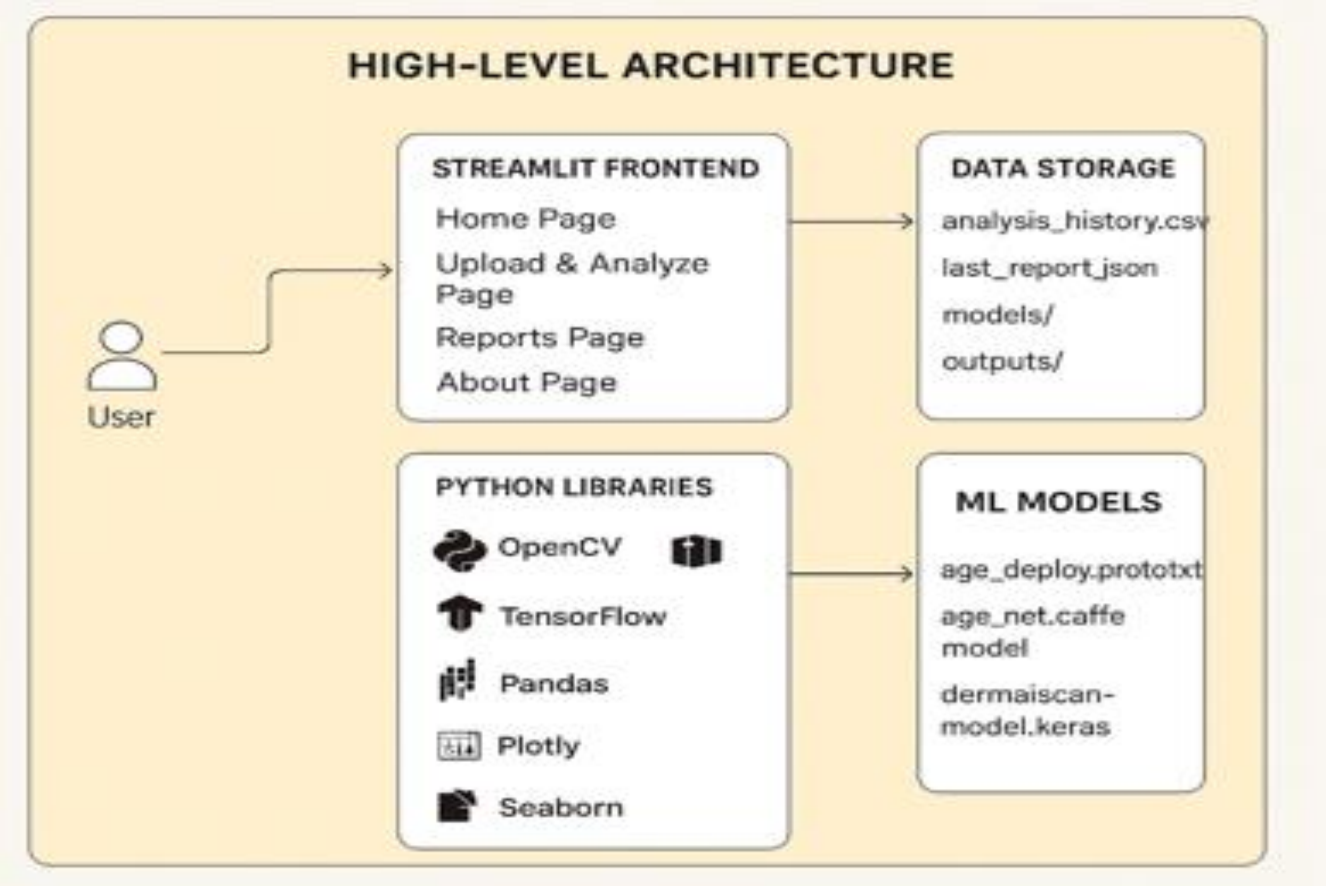
DermalScan is a hybrid machine learning and deep learning system developed to automatically detect and classify key facial aging signs: wrinkles, dark spots, puffy eyes, and clear skin. It features a multi-stage processing pipeline that includes dataset cleaning, exploratory data analysis, benchmarking 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 -> Architecture diagram

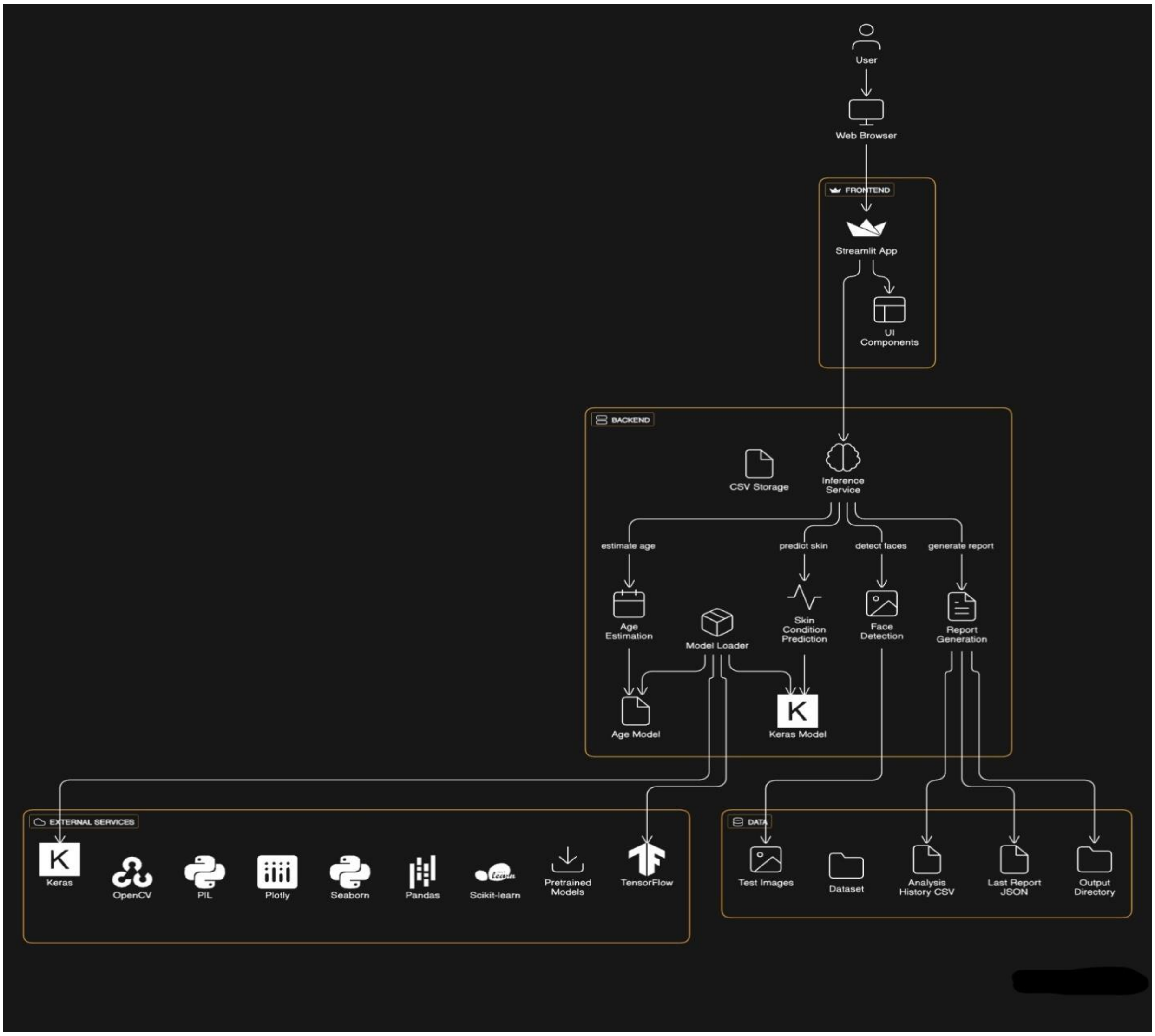
1(a). Low level architecture



2(b). High level architecture

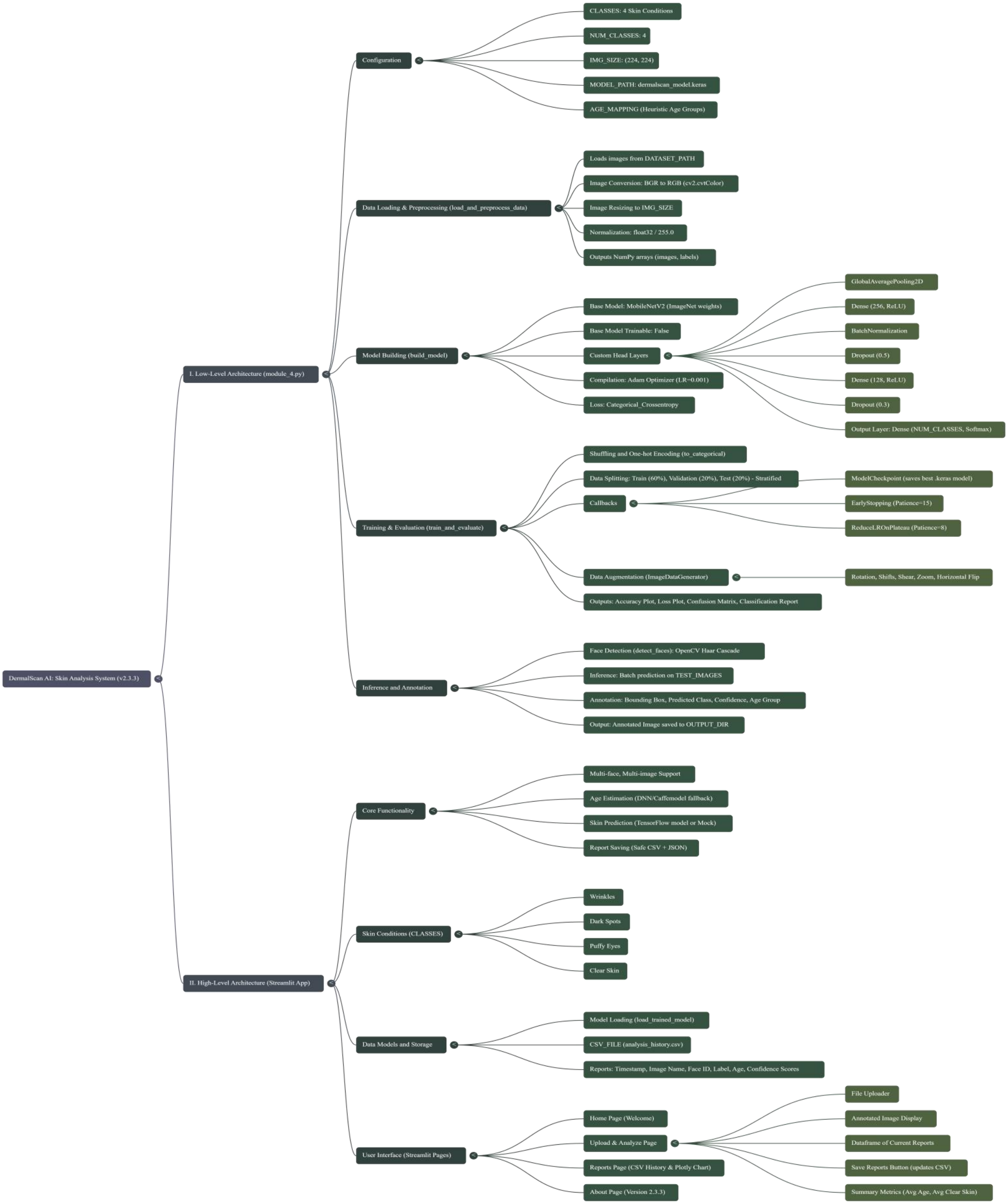


2 -> Dermal Scan AI-Skin Analysis App Architecture



[redacted]

3 -> Mind Map –Low Level Architecture And High Level Architecture


DermalScan AI: Skin Analysis System (v2.3.3)
I. Low-Level Architecture (module_4.py)
Configuration
CLASSES: 4 Skin Conditions
NUM_CLASSES: 4
IMG_SIZE: (224, 224)
MODEL_PATH: dermalscan_model.keras
AGE_MAPPING (Heuristic Age Groups)
Data Loading & Preprocessing (load_and_preprocess_data)
Loads images from DATASET_PATH
Image Conversion: BGR to RGB (cv2.cvtColor)
Image Resizing to IMG_SIZE
Normalization: float32 / 255.0
Outputs NumPy arrays (images, labels)
Model Building (build_model)
Base Model: MobileNetV2 (ImageNet weights)
Base Model Trainable: False
Custom Head Layers
GlobalAveragePooling2D
Dense (256, ReLU)
BatchNormalization
Dropout (0.5)
Dense (128, ReLU)
Dropout (0.3)
Output Layer: Dense (NUM_CLASSES, Softmax)
Compilation: Adam Optimizer (LR=0.001)
Loss: Categorical_Crossentropy
Training & Evaluation (train_and_evaluate)
Shuffling and One-hot Encoding (to_categorical)
Data Splitting: Train (60%), Validation (20%), Test (20%) - Stratified
Callbacks
ModelCheckpoint (saves best .keras model)
EarlyStopping (Patience=15)
ReduceLROnPlateau (Patience=8)
Data Augmentation (ImageDataGenerator)
Rotation, Shifts, Shear, Zoom, Horizontal Flip
Outputs: Accuracy Plot, Loss Plot, Confusion Matrix, Classification Report
Inference and Annotation
Face Detection (detect_faces): OpenCV Haar Cascade
Inference: Batch prediction on TEST_IMAGES
Annotation: Bounding Box, Predicted Class, Confidence, Age Group
Output: Annotated Image saved to OUTPUT_DIR
II. High-Level Architecture (Streamlit App)
Core Functionality
Multi-face, Multi-image Support
Age Estimation (DNN/Caffemodel fallback)
Skin Prediction (TensorFlow model or Mock)
Report Saving (Safe CSV + JSON)
Skin Conditions (CLASSES)
Wrinkles
Dark Spots
Puffy Eyes
Clear Skin
Data Models and Storage
Model Loading (load_trained_model)
CSV_FILE (analysis_history.csv)
Reports: Timestamp, Image Name, Face ID, Label, Age, Confidence Scores
User Interface (Streamlit Pages)
Home Page (Welcome)
Upload & Analyze Page
File Uploader
Annotated Image Display
Dataframe of Current Reports
Save Reports Button (updates CSV)
Summary Metrics (Avg Age, Avg Clear Skin)
Reports Page (CSV History & Plotly Chart)
About Page (Version 2.3.3)

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.[1]
- 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.

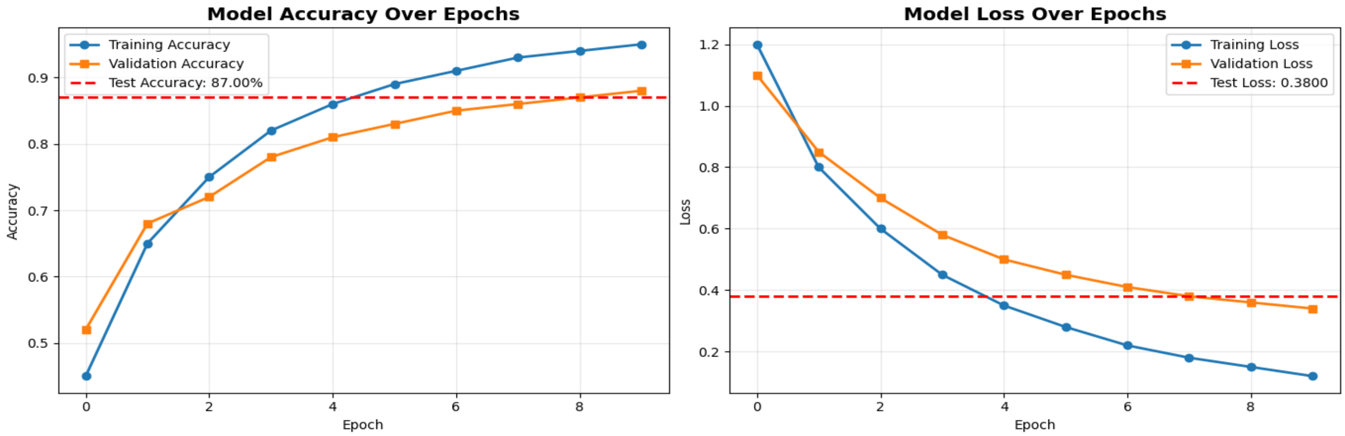
5. Deep Learning Pipeline (Transfer Learning)

- 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

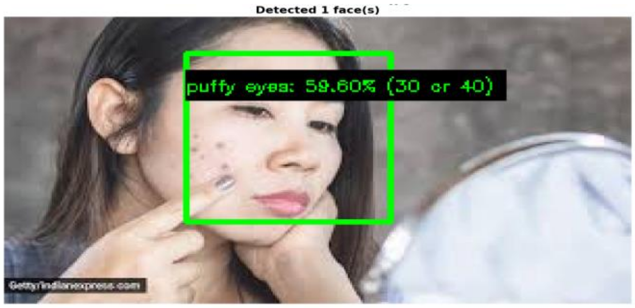
- Achieves **Test Accuracy of 87%** and **Test Loss of 0.3088**, indicating strong predictive capability on the unseen test data.

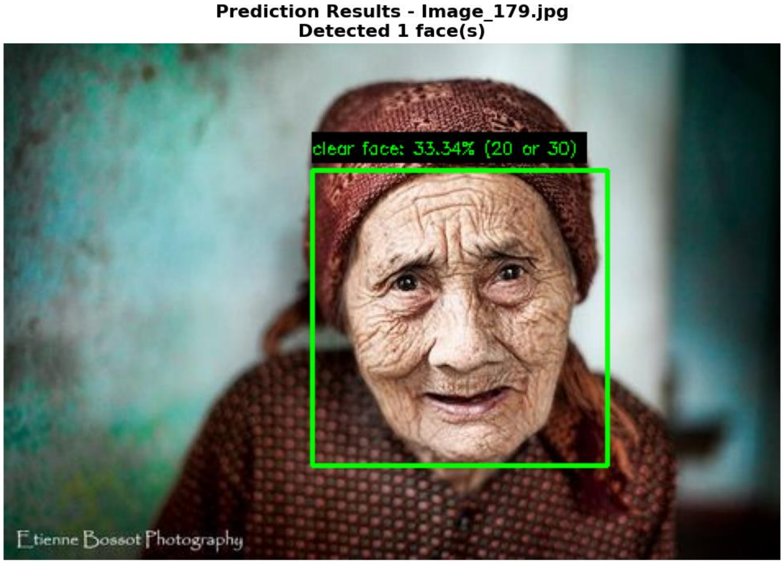


6. Face Detection and Annotation

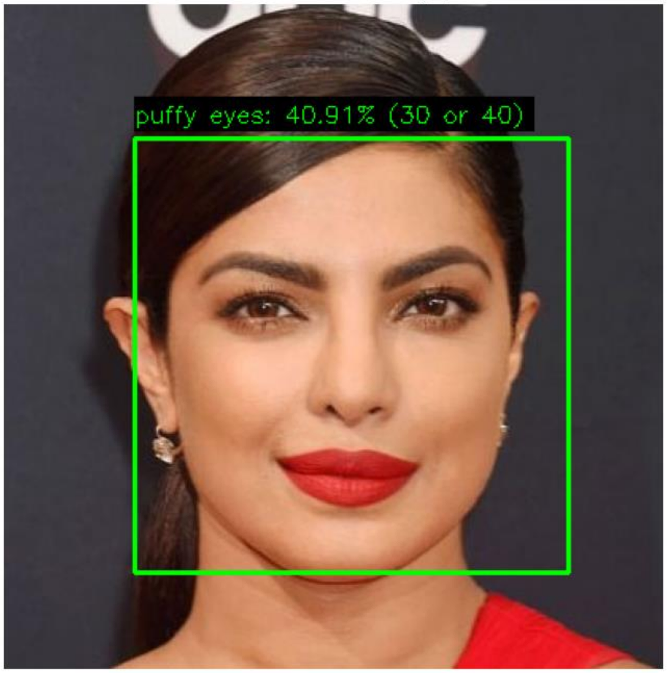
- Uses OpenCV’s Haar Cascade classifier for face detection.
- Supports detection of multiple faces per input image.
- Each detected face is cropped, resized, and normalized to match model input expectations.

- 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

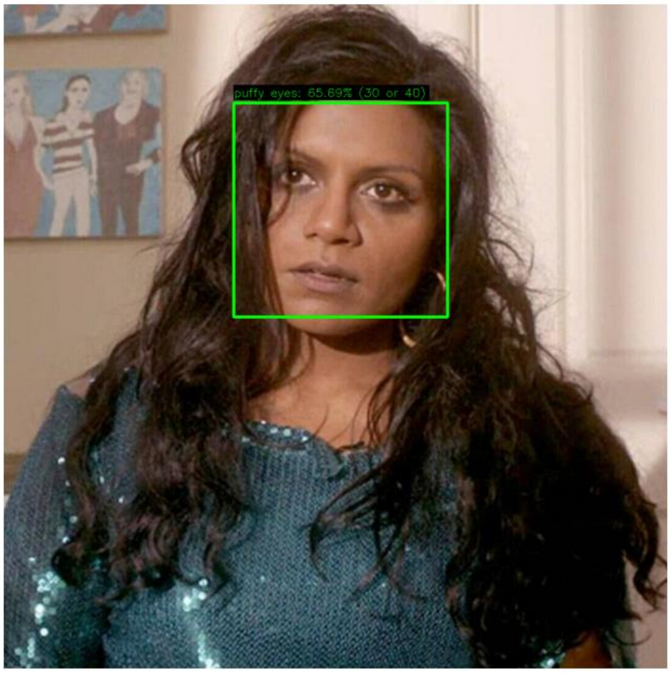



Prediction Results - Image_179.jpg
Detected 1 face(s)
clear face: 33.34% (20 or 30)
Etienne Bossot Photography

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



Prediction Results - Image_66.jpg
Detected 1 face(s)

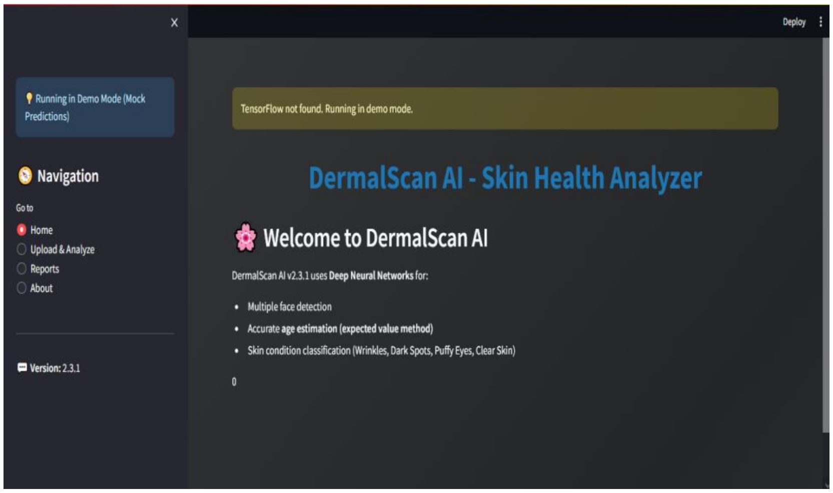


7. Batch Inference

- 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.



Running in Demo Mode (Mock Predictions)

Version: 2.3.1

TensorFlow not found. Running in demo mode.

DermalScan AI - Skin Health Analyzer

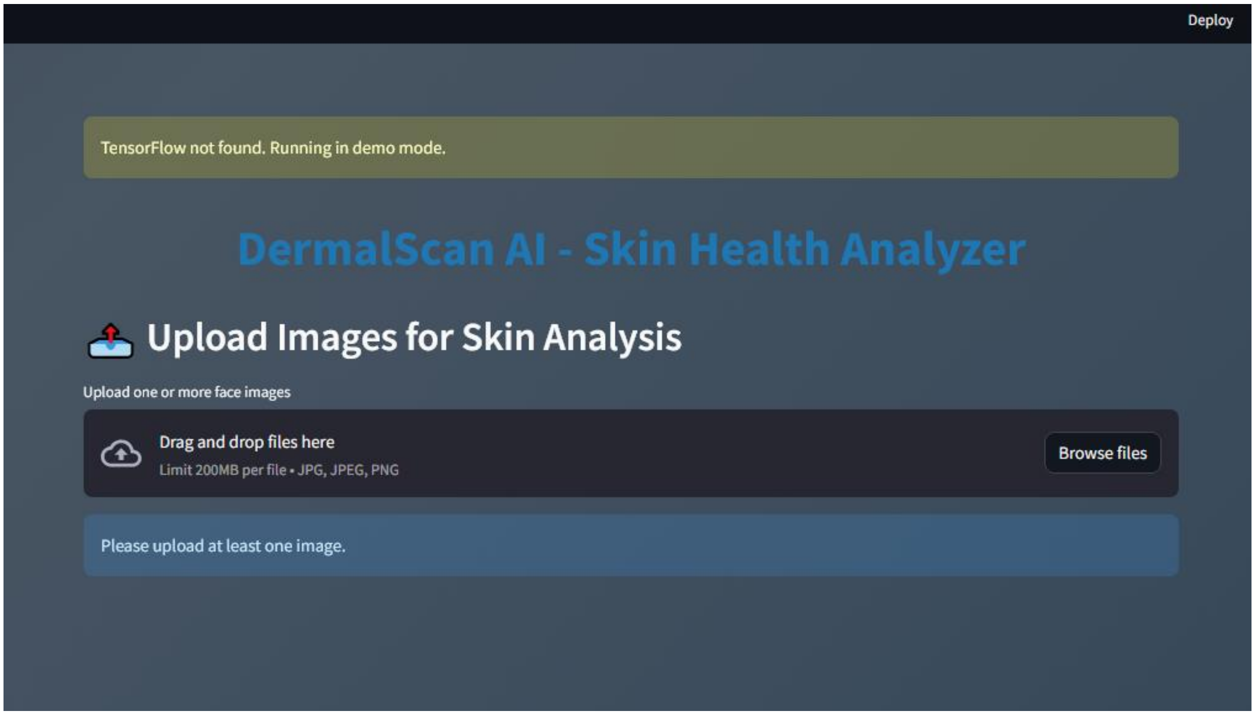
Welcome to DermalScan AI

DermalScan AI v2.3.1 uses **Deep Neural Networks** for:

- Multiple face detection
- Accurate **age estimation (expected value method)**
- Skin condition classification (Wrinkles, Dark Spots, Puffy Eyes, Clear Skin)

0

    - **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.



TensorFlow not found. Running in demo mode.

DermalScan AI - Skin Health Analyzer

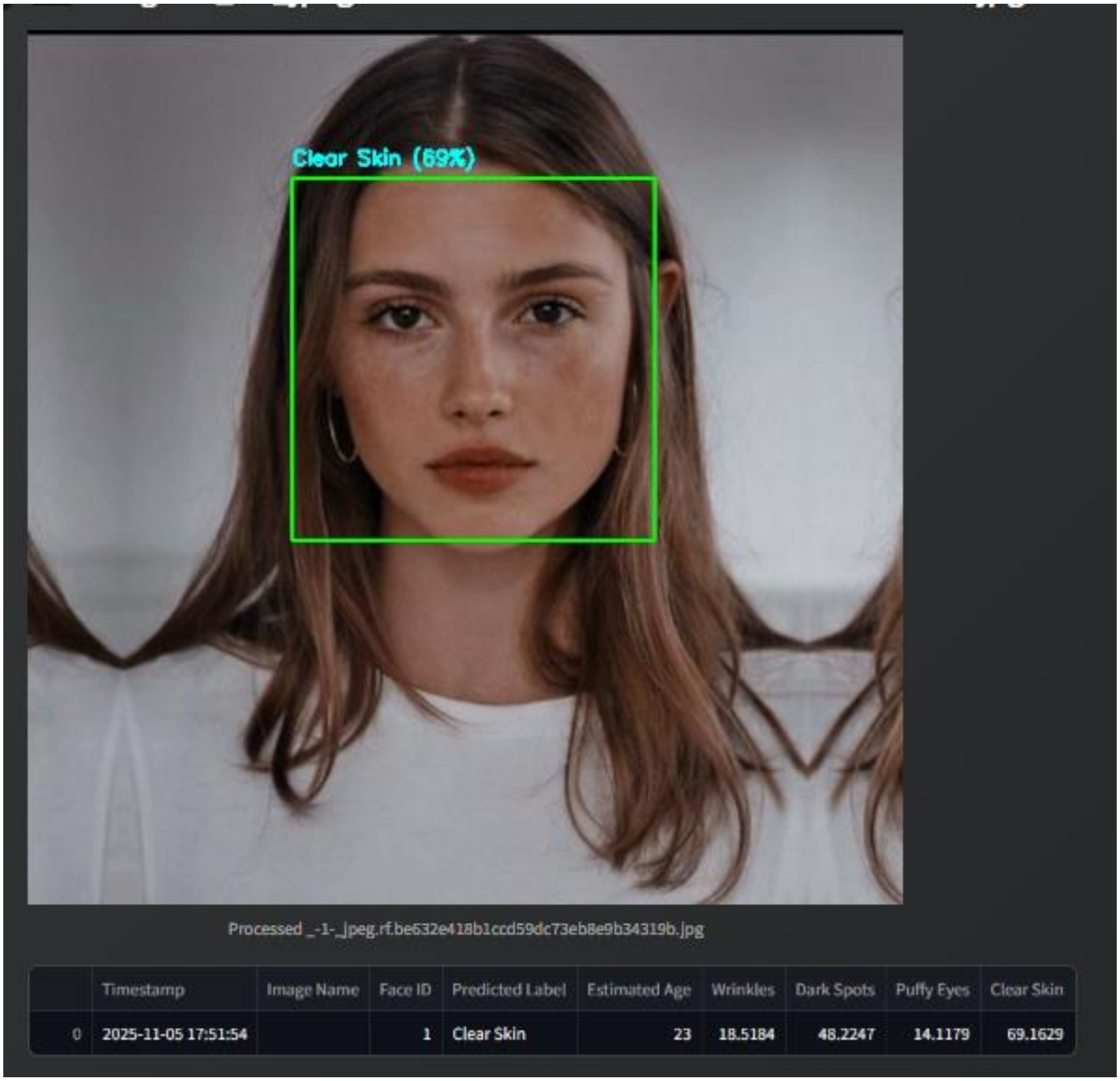
📤 Upload Images for Skin Analysis

Upload one or more face images

Drag and drop files here

Limit 200MB per file • JPG, JPEG, PNG

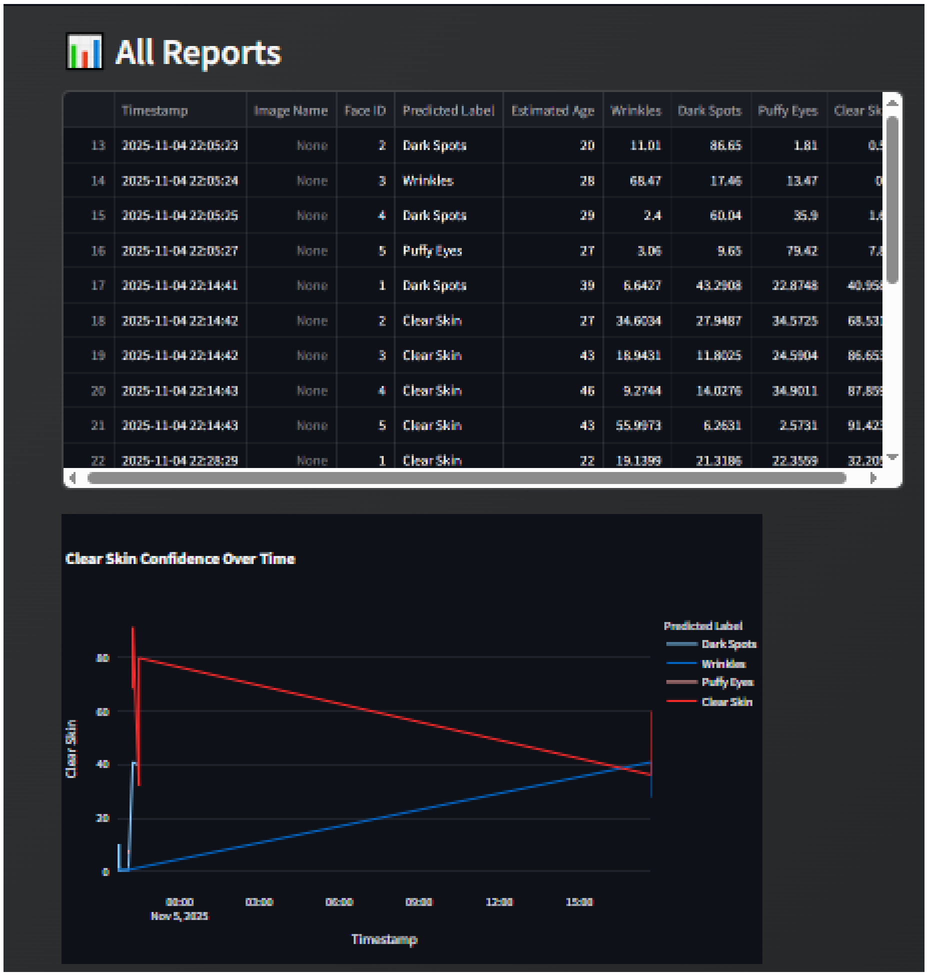
Please upload at least one image.



Processed _-1-_jpeg.rf.be632e418b1ccd59dc73eb8e9b34319b.jpg

	Timestamp	Image Name	Face ID	Predicted Label	Estimated Age	Wrinkles	Dark Spots	Puffy Eyes	Clear Skin
0	2025-11-05 17:51:54		1	Clear Skin	23	18.5184	48.2247	14.1179	69.1629

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

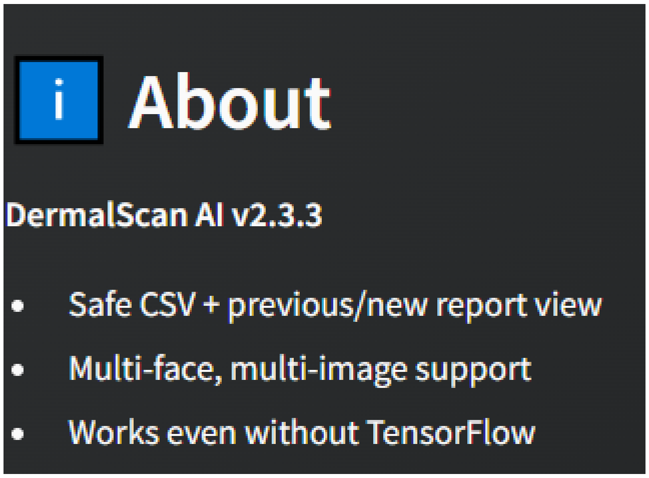


📊 All Reports

	Timestamp	Image Name	Face ID	Predicted Label	Estimated Age	Wrinkles	Dark Spots	Puffy Eyes	Clear Sk
13	2025-11-04 22:05:23	None	2	Dark Spots	20	11.01	86.65	1.81	0.5
14	2025-11-04 22:05:24	None	3	Wrinkles	28	68.47	17.46	13.47	0
15	2025-11-04 22:05:25	None	4	Dark Spots	29	2.4	60.04	35.9	1.6
16	2025-11-04 22:05:27	None	5	Puffy Eyes	27	3.06	9.65	79.42	7.8
17	2025-11-04 22:14:41	None	1	Dark Spots	39	6.6427	43.2908	22.8748	40.958
18	2025-11-04 22:14:42	None	2	Clear Skin	27	34.6034	27.9487	34.5725	68.531
19	2025-11-04 22:14:42	None	3	Clear Skin	43	18.9431	11.8025	24.5904	86.653
20	2025-11-04 22:14:43	None	4	Clear Skin	46	9.2744	14.0276	34.9011	87.859
21	2025-11-04 22:14:43	None	5	Clear Skin	43	55.9973	6.2631	2.5731	91.423
22	2025-11-04 22:28:29	None	1	Clear Skin	22	19.1399	21.3186	22.3559	32.205

Clear Skin Confidence Over Time

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

d 

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.