Dermal Scan
AI Facial Skin Aging Detection App

Presented by
D.Upendar

Identify aging features and provide accurate predictions.

Agenda

Project Statement

- An overview of the project's goal: to build an AI system for detecting facial aging signs using MobileNetV2 and a Streamlit web app.

Objectives

- Identify aging features and provide accurate predictions.including model training, UI development, and the key deliverables of the project.

Methodology

- A technical deep dive into the data preprocessing, model architecture, and the end-to-end analysis pipeline from image upload to result.

Annotated outputs, accuracy, and performance evaluation.

Results

- Annotated outputs, accuracy, and performance evaluation.

Future Enhancements

- A discussion of the project's potential roadmap, including plans for dataset expansion and mobile integration.

Conclusion

- A summary of the project's achievements and the key learnings from the development process.

Project Statement

Objective: Develop a deep learning-based system to detect and classify facial aging signs from user-uploaded images.

Core Technology:

- **Model:** MobileNetV2 for high-efficiency image classification.
- **Application:** An interactive web UI built with Streamlit.
- **Backend:** Python with OpenCV for image processing and analysis.

Objectives & Outcomes

Key Objectives:

Develop a Detection Pipeline: Build an end-to-end system to process an image, detect a face using Haar Cascades, and prepare it for model inference.

Train the Classification Model: Fine-tune a pre-trained **MobileNetV2** model to accurately classify six categories of skin conditions, including acne, wrinkles, and dark spots.

Build an Interactive UI: Create a user-friendly web interface with **Streamlit** that allows users to easily upload an image and view the results.

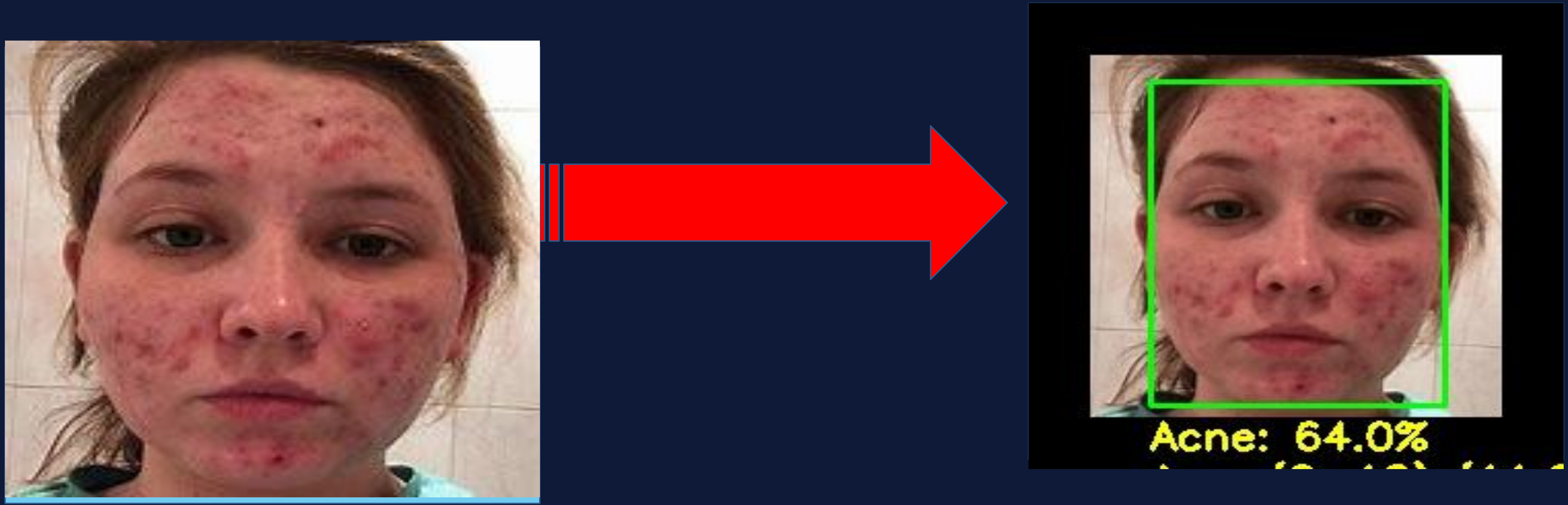
Provide Detailed Results: Display the classification result with a confidence score and an estimated age, annotated directly onto the output image.

Project Outcomes

A functional web application that delivers real-time facial analysis with a processing time of **under 5 seconds**.

A highly accurate classification model with a validation accuracy of **99.75%**.
Features to **download** the annotated image and a CSV log of the prediction details.

Example-1



Example -2

Tech Stack

Python Version
python==3.10
Core
numpy>=1.23.0
pandas>=1.5.0

Machine Learning / Deep Learning
tensorflow>=2.10.0
scikit-learn>=1.2.0

Computer Vision
opencv-python>=4.7.0

Web UI
streamlit>=1.24.0

Visualization
matplotlib>=3.7.0

Image Ops:OpenCV, NumPy, Haarcascade

Dataset & Preprocessing

Collected a diverse facial image dataset from Kaggle, including categories such as wrinkles, dark spots, puffy eyes, scars, and clear skin.

Ensured balanced distribution and quality samples for effective model training and evaluation.

- Total Images: **21,510** across **6 classes**
- Classes: Acne, Clear Face, Dark Spots, Puffy Eyes, Scars, Wrinkles
- All images resized to **224×224** pixels for model input

Data Preprocessing

- The goal of this pipeline was to transform the raw image collection into a normalized, expanded, and memory-efficient dataset ready for training a **MobileNetV2** model.

- **Resizing** → All images are resized to a uniform size of **224 × 224 pixels**.

- **Normalization** → Pixel values are scaled between **0 and 1** (`rescale=1./255`).

- **Folder Structure** → Images are organized into class-wise folders for training.

- **Consistency** → Original resized images are also saved to the processed dataset.

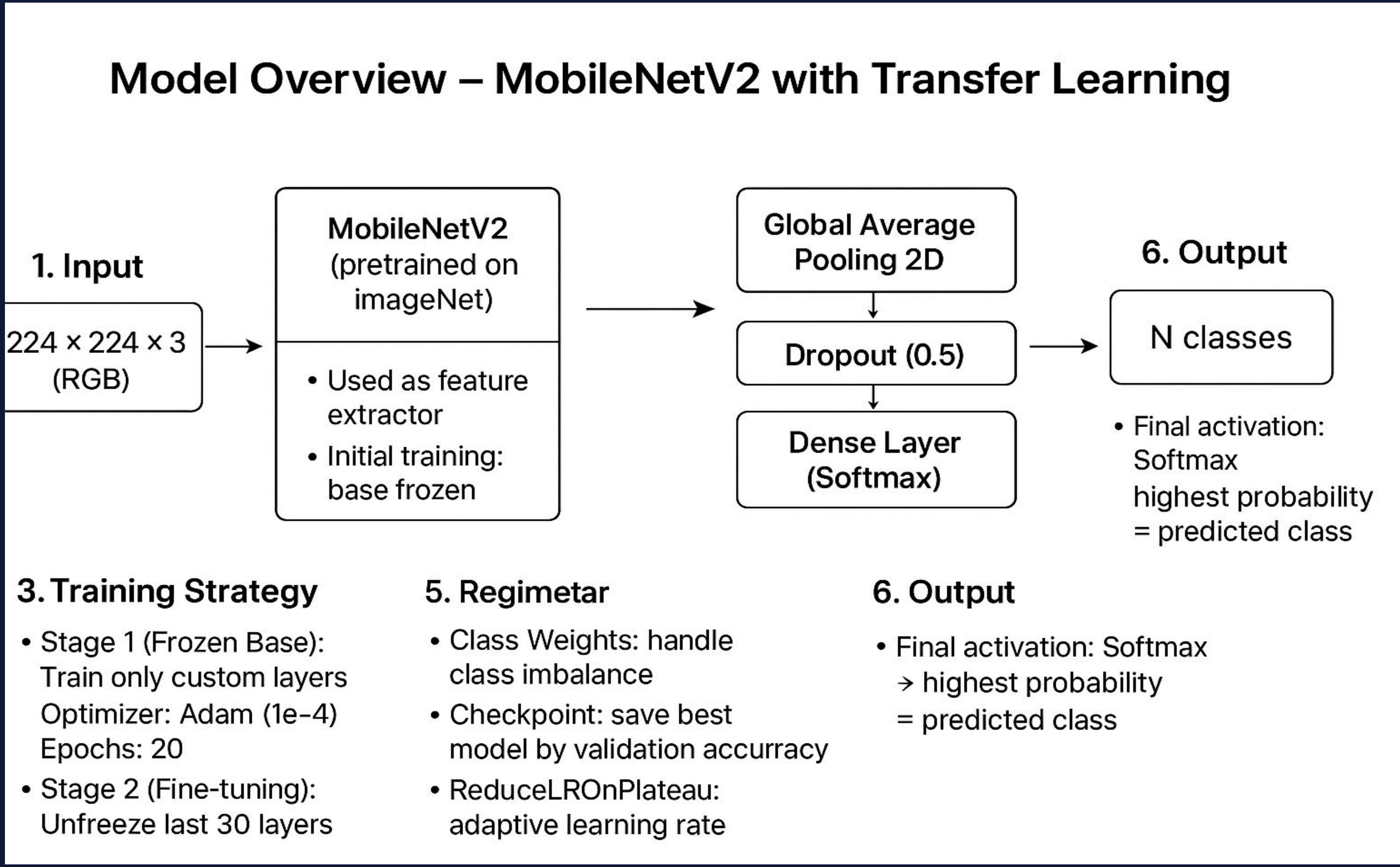
Data Augmentation

To improve model generalization and prevent overfitting, each image is augmented using:

- **Rotation (±25°)** – helps learn orientation invariance.
- **Width & Height Shift (±15%)** – makes the model robust to positional changes.
- **Shearing (10%)** – introduces geometric transformations.
- **Zoom (±20%)** – mimics different camera distances.
- **Augmentation Strategy: 4 augmented versions** were generated for every original image (`augment_per_image = 4`). The transformations included:**Geometric:** Random rotation ($\pm 25^{\circ}$), shifts (±15%), shear, zoom, and **horizontal flipping**.
- **Brightness Adjustment (0.8–1.2)** – simulates lighting variations.
- **Fill Mode = Nearest** – fills missing pixels after transformations.

Model Overview

We employed a **Transfer Learning** approach using the **MobileNetV2** architecture, fine-tuning it in two distinct stages to achieve optimal performance on the skin condition classification task.



Training Strategy

Stage 1 – Frozen Base Training

- Base **MobileNetV2** kept **frozen** (pretrained ImageNet features).
- Only the **custom classifier head** is trained.
- **LR = 1e-4**, **Epochs = 20**, Batch Size = 32.
- **Purpose:** Quickly learn how to map general features → skin condition classes.

Stage 2 – Fine-Tuning

- Unfreeze the **last 30 layers** of MobileNetV2.
- Retrain both base + classifier with a **lower LR (1e-5)**.
- **Epochs = 50**, Batch Size = 32.
- **Purpose:** Adapt pretrained features to **domain-specific patterns** (wrinkles, scars, acne, etc.).

- **Optimizer:** Adam
- **Loss Function:** Categorical Crossentropy (multi-class)
- **Mtric:** Accuracy

Two-Stage Fine-Tuning Process

Stage	Goal	Layers Trained	Learning Rate	Epocs
Stage 1: Future Extraction	Quickly train the new classification head.	**MobileNetV2 base was frozen**. Only the custm **Global Average Pooling,Doropout, and Dense** layers were trained.	1×10−4 (Initial)	20 (with early stopping)
Stage 2: Fine-Tuning	Tune the base network to the specific dermal feature set.	The custom head **PLUS the last 30 layers** of the MobileNetV2 base were **unfrozen** and trained.	1×10−5 (10x smaller)	50 (with early stopping)

Training Safeguards(Callbacks)

We employed robust callbacks to manage the training process dynamically and ensure we captured the best possible model performance:

ModelCheckpoint:

- Saved weights **only when validation accuracy improved**
- Ensured we captured the **best-performing model**
- Final output: **`mobilenetv2_best_model.h5`**

ReduceLROnPlateau:

- Monitored **validation loss**
- Reduced learning rate by **50%** if no improvement for **3 epochs**
- Helped the model escape **local minima**

EarlyStopping:

- Stopped training if **validation loss** did not improve for **5 epochs**
- **Restored best weights** to prevent overfitting and save time

Accuracy

Overall Performance Metrics:

Metric	Result	Interpretation
Validation Accuracy	99.75%	The model correctly classified 99.75% of the skin condition images in the test set.
Loss	0.0068	An extremely low loss value, indicating high confidence in the correct predictions.

EvaluationResults

Evaluation Results

1. Model Performance

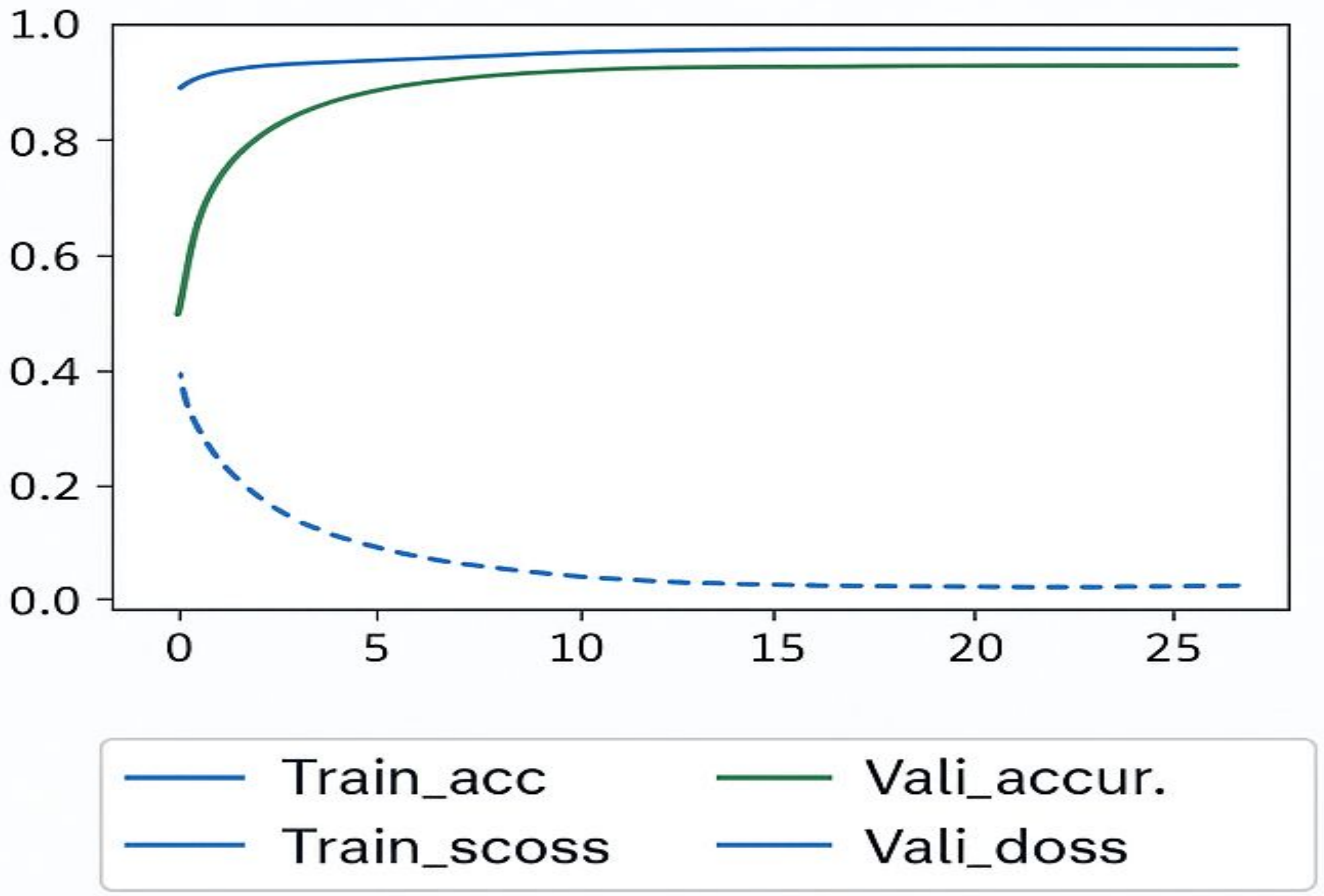
Validation Accuracy	99.75 %
Validation Loss	0.0068

Best model: mobilenetv2_best_model.h5

2. Classification Report (per class)

	precision	recall	f1-score	support
acne	1.00	1.00	1.00	862
clear_face	1.00	1.00	1.00	283
dark_spots	0.99	1.00	0.99	412
puffy_eyes	1.00	1.00	1.00	359
scars	1.00	1.00	1.00	602
wrinkles	1.00	1.00	1.00	709
macro avg	1.00	1.00	1.00	3227

3. Accuracy & Loss Curves



Prediction Pipeline

Here are the models explicitly included in the pipeline.

Skin Condition Classification Model

- **Model Name: MobileNetV2** (Pre-trained and Fine-tuned)
- **Purpose:** Classifies input images into one of six skin conditions.
- **File (Weights):** `mobilenetv2_best_model.h5`
- **File (Architecture):** The Python Keras code defines the architecture.

Age Prediction Model

This model uses the **Caffe framework**, suggesting it's likely a pre-existing model for general age estimation.

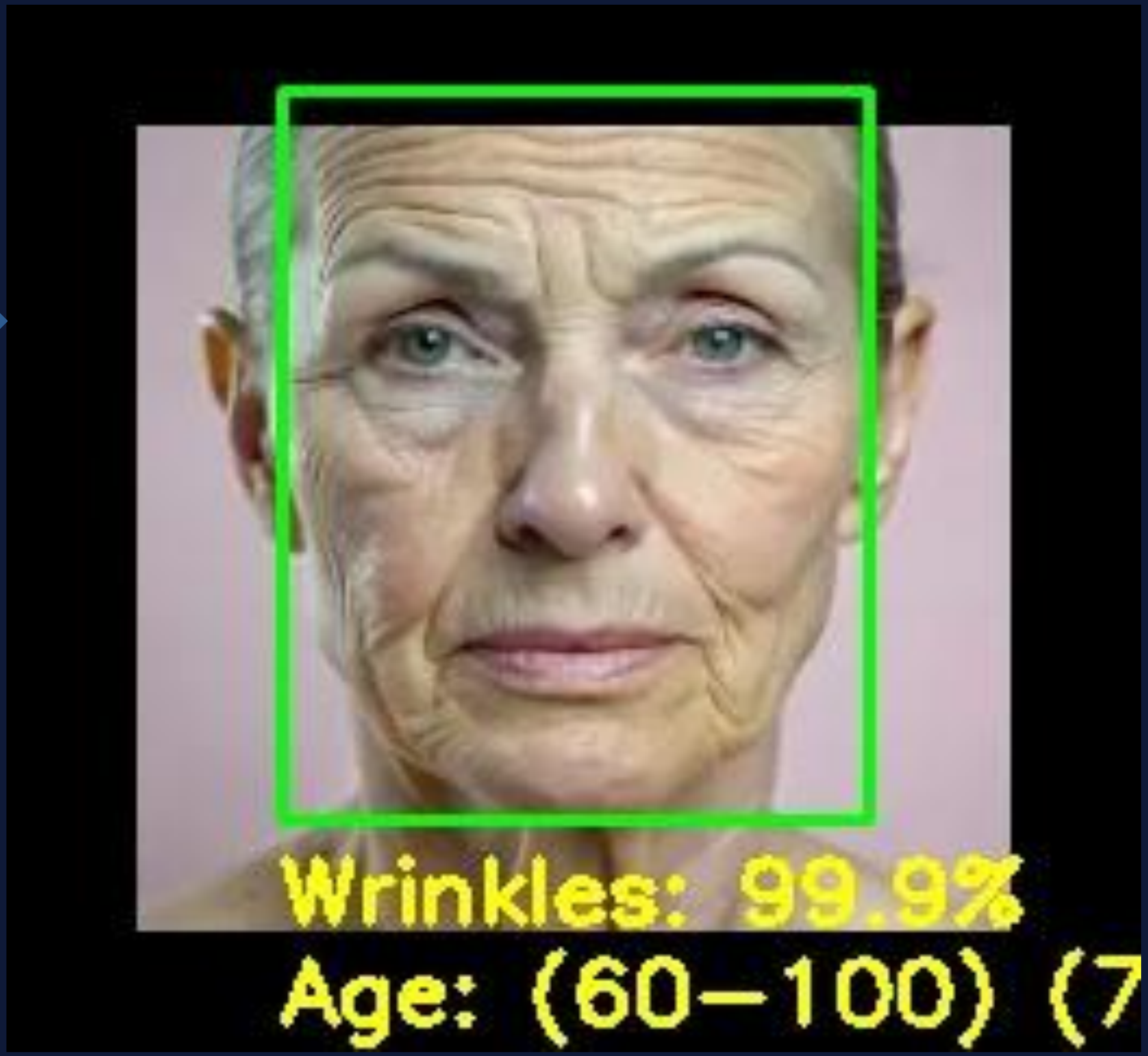
- **Model Type:** Caffe Model (Deep Learning)
- **Model Weights:** `age_net.caffemodel`
- **Model Architecture:** `age_deploy.prototxt`
- **Purpose:** To estimate the age of the detected face.

Face Detection Model (External Component)

This component is a separate, highly efficient deep learning face detector (often SSD or similar), used as an alternative or complement to the OpenCV Haar Cascades for more robust face localization.

- **Model Type:** Deep Learning Face Detector (likely Single Shot Detector - SSD)
- **Model Weights:** `opencv_face_detector_uint8.pb` (TensorFlow/Protobuf file)
- **Model Architecture:** `opencv_face_detector.pbtxt` (Protobuf text file)
- **Purpose:** Accurately detect the location of a face in an image.

Prediction Images



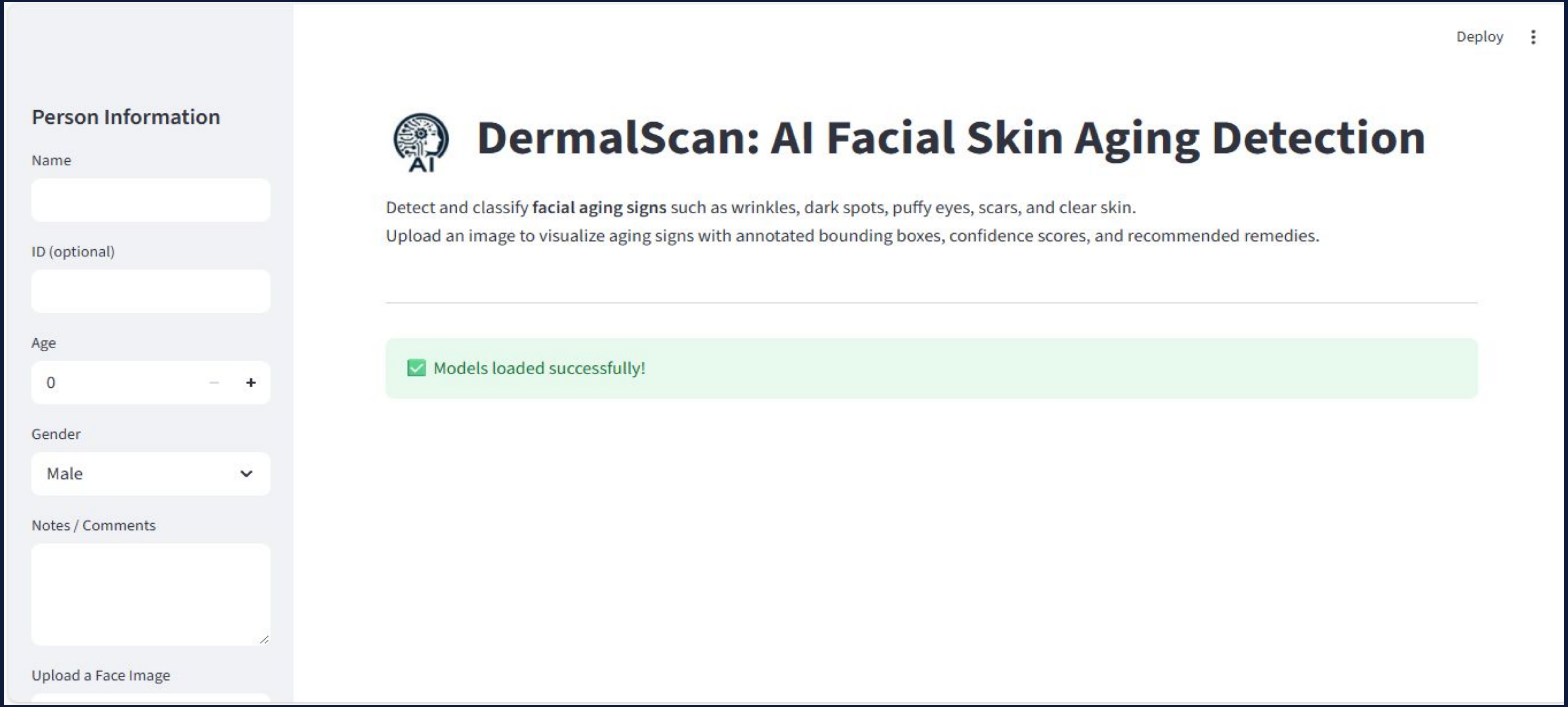
Deployment: DermalScan Web Application (Streamlit)

User Input: The interface supports drag-and-drop or file browsing for image submission.

- **File Type Limit:** The app accepts standard image formats like **JPG, JP(E)G, and PNG**.
- **File Size Limit:** The maximum file size is restricted to **200MB**.

Visual Output: The primary output is the visualization of aging signs with **annotated bounding boxes and confidence scores**. This is a crucial feature, showing *where* the model detected the condition on the face.

Streamlit UI



Prediction Details

Deploy

Upload Image

Choose an image...

Drag and drop file here

Limit 200MB per file • JPG, JP...

Browse files

uppu.jpg

55.2KB

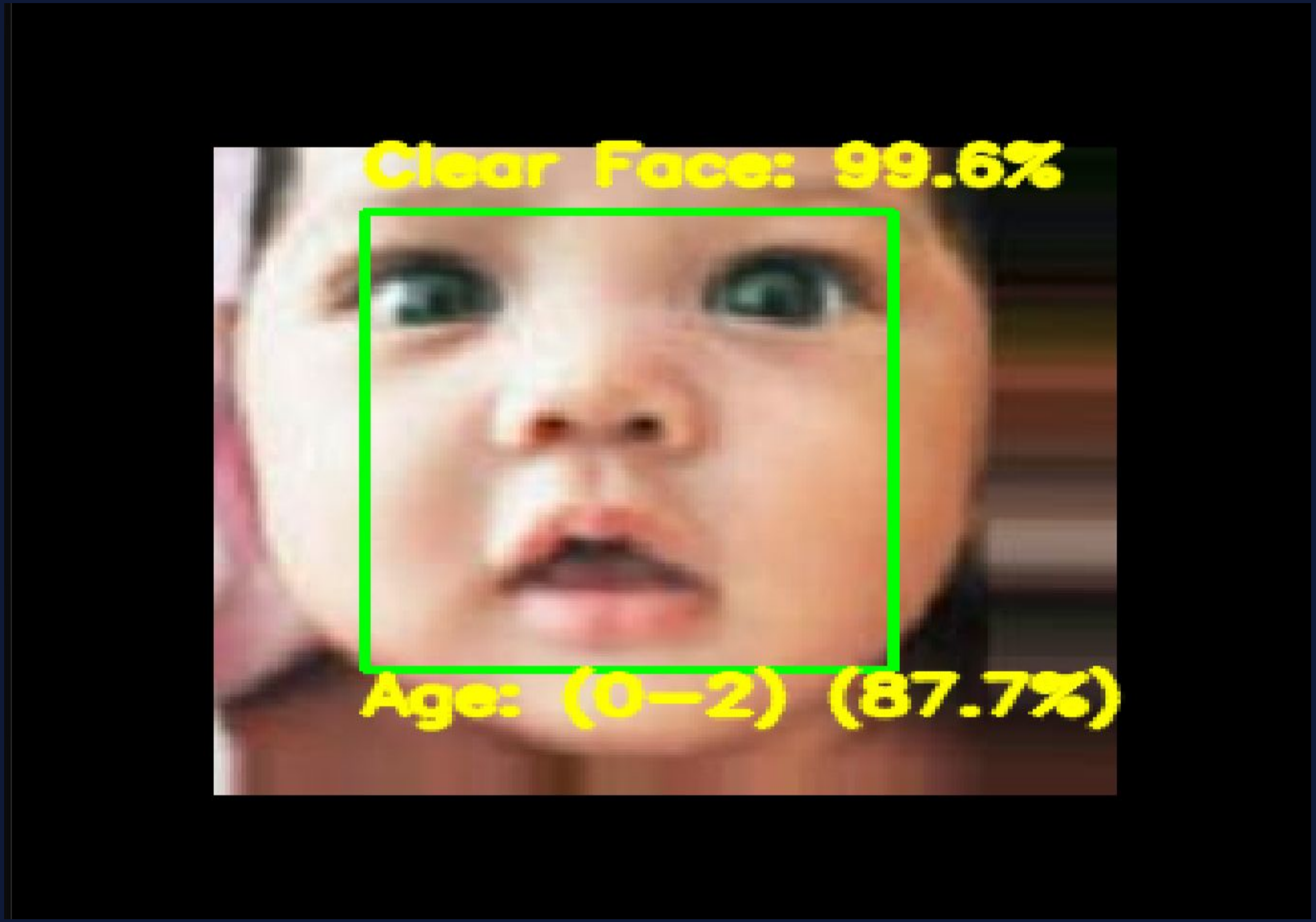
	value
Dermal Class	Clear Face
Confidence	88.8%
Predicted Age	(15-20)
Age (Years)	22.4
Age Confidence	53.5%
Bounding Box	132,245,305,476

Download Prediction Details (CSV)

Annotated image saved at: C:\Dermal scan\output\annotated_uploaded_image.jpg

Download Annotated Image

otated Image



Evaluation Metrices

```
(dermal_env) C:\Dermal scan>python evaluate_model.py
2025-09-12 15:07:49.672257: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results du
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-09-12 15:07:55.002137: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results du
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
📁 Loading dataset...
✅ Dataset loaded: (21510, 224, 224, 3), (21510, 6)
📁 Loading trained model...
2025-09-12 15:10:11.551655: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructio
ns in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or eval
uate the model.
✅ Loaded model from C:\Dermal scan\mobilenetv2_best_model.h5
101/101 ━━━━━━━━━━━━━━━━━━━━ 62s 545ms/step - accuracy: 0.9975 - loss: 0.0068
📊 Validation Accuracy: 0.9975, Loss: 0.0068
101/101 ━━━━━━━━━━━━━━━━━━━━ 61s 584ms/step

📄 Classification Report:
              precision    recall  f1-score   support

        acne       1.00      1.00      1.00       862
  clear_face       1.00      1.00      1.00       283
  dark_spots       0.99      1.00      0.99       412
  puffy_eyes       0.99      1.00      1.00       359
       scars       1.00      1.00      1.00       602
    wrinkles       1.00      0.99      1.00       709

    accuracy                           1.00      3227
   macro avg       1.00      1.00      1.00      3227
weighted avg       1.00      1.00      1.00      3227
```

Challenges Faced

Dataset Collection & Preprocessing

- Limited availability of **high-quality, balanced skin condition images**.
- Required **augmentation (rotation, zoom, brightness, etc.)** to increase dataset diversity.

Model Training

- Risk of **overfitting** due to small dataset.
- Solved by **transfer learning (MobileNetV2)**, **dropout**, and **callbacks (EarlyStopping, ReduceLROnPlateau, ModelCheckpoint)**.
  UI & Integration

Initial Attempt – EfficientNetB0

- Started with **EfficientNetB0** as the base model.
- Faced **longer training times** and **higher computational demand**.
- Although accuracy was good, it was **too heavy for real-time deployment**.

2. Transition to MobileNetV2

- Switched to **MobileNetV2**, a more **lightweight & efficient** model.
- Achieved **faster training**, **lower inference time**, and **reduced memory usage**.
- Maintained **high accuracy (~99%)** while being **deployment-friendly**.

Future Enhancements

- Larger dataset
- Multi-class aging signs
- Mobile app integration
- Real-time detection

Key Use Cases for the DermalScan Application

🧑‍⚕️ Clinical & Aesthetic Use

Pre-screening & Triage: Provides dermatologists with a quantified skin report before consultation.

reatment Planning: Uses bounding boxes and percentage scores for planning laser treatments or fillers.

Measurable Results: Offers verifiable proof of treatment success (before/after data).

Consumer & Skincare

Personalized Recommendations: Identifies precise skin needs (Wrinkles, Dark Spots, Acne) for product matching.

Routine Tracking: Users can monitor the quantifiable effectiveness of their skincare routine over time.

Age-Based Validation: Uses precise age estimation to deliver highly targeted advice.

Objective Efficacy Testing: Measures the success of clinical tria
ls by tracking reduction in aging signs (e.g., from 85% wrinkles to 60%).
Data-Driven R&D: Analyzes large datasets to identify market trends and target product development.

Quantifiable Benchmarks: Establishes objective benchmarks for product claims.

Conclusion

The DermalScan project successfully validated a highly accurate deep learning system for classifying facial skin conditions, confirming the viability of using efficient models

By employing a **Two-Stage Transfer Learning strategy** with **MobileNetV2**, and utilizing robust data handling through **Data Augmentation** and **Balanced Class Weighting**, we achieved exceptional performance: **99.75% Validation Accuracy** and a **perfect 1.00 F1-Score**.

The final comprehensive system integrates a classifier, a face detector, and an age model, all deployed as an interactive **Streamlit web application**. This provides a practical, efficient, and highly accurate tool ready for non-invasive skin health screening.

Thank you!

Thank you for your time.

D.Upendar
upendardduggineni24@gmail.com
GitHub:https://github.com/SpringBoardMentor0781j/Dermal_Scan_Project_2025.git