Gender & Age Detection App

Overview

This application predicts the gender and age of a person from an uploaded image using a FastAPI backend and a Streamlit frontend. It leverages Google's Gemini Pro Vision model for inference and ensures high confidence in predictions.

Features

- Upload a face image
- Predict **Gender**, **Age**, and **Confidence** level
- Only return predictions if confidence is 90% or higher
- View and append results in a session table
- Download prediction history as a CSV file

Tech Stack

Backend

- **Framework:** FastAPI
- **Image Processing:** Pillow
- **Model:** Gemini Pro Vision via `google.generativeai`
- **Deployment:** Render

Frontend

- **Framework:** Streamlit
- **Features:**
- File uploader
- Spinner while loading
- Displays prediction
- Session log table
- CSV download button
- **Deployment:** Streamlit Cloud

Backend Endpoint

POST `/Analyze_image`

- Accepts: `multipart/form-data` with a file

- Returns:

```
{
  "response": {
    "Gender": "Male",
    "Age": 20,
    "Confidence": "91%"
  }
}
```

- If no face or low confidence:

```
{
  "response": {
    "error": "Confidence below 90%. Cannot provide a reliable prediction."
  }
}
```

Prompt for Gemini Model

```
You are a facial analysis expert.
Your task is to predict the gender and age of the person in the uploaded image.

Only respond if you are at least 90% confident in your prediction.

If no human face is detected:
{
  "error": "No human detected."
}

If confidence is below 90%:
{
  "error": "Confidence below 90%. Cannot provide a reliable prediction."
}

If confident (>=90%):
{
  "Gender": "Male/Female/Transgender",
  "Age": integer,
  "Confidence": "percent string like '91%'"
}

Return ONLY valid JSON. No markdown, no backticks, no extra text.
```

Future Improvements

- Face bounding box overlay
- Support for multiple faces
- Integration with DeepFace or local models for offline use
- Confidence tuning via frontend slider

Maintainer

Waris Hayat\ GitHub Repo