Advanced Face Detection System

# Overview

This project implements a real-time face detection system using OpenCV, age, and gender classification models. It also includes additional functionality such as eye detection toggling, face image capture, and real-time FPS display.

The key features of this project include:

- Face detection using Haar cascades.

- Age and gender prediction using pre-trained deep learning models.

- Toggle to enable or disable eye detection.

- Capturing and saving face images locally.

- User interface with buttons for various functionalities.

- Real-time FPS display.

# Requirements

Before running the project, ensure you have the following prerequisites:

## Dependencies:

Python 3.x  
OpenCV  
NumPy

## Python Libraries:

You can install the necessary libraries using `pip`:

```bash  
pip install opencv-python opencv-python-headless numpy  
```

## Models:

Haarcascade Frontal Face and Eye detection models, which come pre-installed with OpenCV.  
Pre-trained Caffe models for age and gender classification (Make sure you download and provide the paths for the models as shown below).

# Folder Structure

The folder structure for this project should look like this:

```bash  
AdvancedFaceDetectionSystem/  
│  
├── age\_net.caffemodel  
├── deploy\_age.prototxt  
├── gender\_net.caffemodel  
├── deploy\_gender.prototxt  
├── face\_detection.py  
└── captured\_faces/ (this folder will be created automatically)  
```

# How to Run

1. Clone the repository or download the project files to your local machine.

2. Install the required dependencies using the command:

```bash  
pip install opencv-python numpy  
```

3. Download the pre-trained models for age and gender and place them in the project directory:

- [Age Caffe Model](https://github.com/yu4u/age-gender-estimation/raw/master/age\_net.caffemodel)  
- [Gender Caffe Model](https://github.com/yu4u/age-gender-estimation/raw/master/gender\_net.caffemodel)  
- [Deploy Age Prototxt](https://github.com/yu4u/age-gender-estimation/raw/master/deploy\_age.prototxt)  
- [Deploy Gender Prototxt](https://github.com/yu4u/age-gender-estimation/raw/master/deploy\_gender.prototxt)

4. Run the script:

```bash  
python face\_detection.py  
```

5. User Interface: A window will appear showing real-time video from your webcam, with the following options:  
- \*\*Toggle Detection\*\*: Turn face detection on/off.  
- \*\*Toggle Eyes\*\*: Turn eye detection on/off.  
- \*\*Capture Face\*\*: Capture and save the detected face to the `captured\_faces/` folder.  
- \*\*Exit\*\*: Close the application.

6. Keyboard Command: Press `q` to quit the application.

# Project Functionality

### 1. Face Detection  
The system detects faces using OpenCV's pre-trained `haarcascade\_frontalface\_default.xml` model. A green rectangle is drawn around the detected faces.

### 2. Eye Detection (Toggle)  
Users can toggle the eye detection feature using the "Toggle Eyes" button. When enabled, blue rectangles will appear around the detected eyes within the face region.

### 3. Age and Gender Prediction  
The system estimates the gender and age group of the detected faces using pre-trained Caffe models. The predictions are displayed on the frame above the detected face.

### 4. Capture Face  
By clicking the "Capture Face" button, the first detected face is saved as an image in the `captured\_faces/` folder with a timestamp as the filename.

### 5. FPS Display  
The frame per second (FPS) rate is calculated and displayed in the top-left corner of the window, providing information about the performance of the system.

# Known Issues

- \*\*Camera Not Detected\*\*: Ensure your webcam is properly connected. If you're using an external webcam, try specifying the correct camera index in `cap = cv2.VideoCapture(0)` (use `1` for an external webcam).

- \*\*Low FPS\*\*: On some machines, processing might be slower depending on the CPU/GPU. Consider optimizing by reducing image resolution or processing fewer frames per second.

# Future Enhancements

- Add functionality for multi-face detection and capture.  
- Optimize model inference for better performance on low-end devices.  
- Add emotion detection and facial landmarks tracking.

# License

This project is open-source and free to use. No license is provided.