Face Detection Project 🎯

A Python-based face detection application using OpenCV that can detect faces in images and provide real-time face detection through webcam.

Features ✨

- **Image Face Detection**: Upload any image and detect faces with bounding boxes
- **Real-time Detection**: Use your webcam for live face detection
- **Eye Detection**: Also detects eyes within detected faces
- **Screenshot Capture**: Save screenshots during real-time detection
- **Multiple Output Formats**: Save processed images with timestamps

Requirements 📋

- Python 3.7 or higher
- OpenCV (cv2)
- NumPy
- A working webcam (for real-time detection)

Installation 🚀

1. **Clone the repository**

   ```bash
   git clone https://github.com/yourusername/face-detection-project.git
   cd face-detection-project
   ```

2. **Create a virtual environment (recommended)**

   ```bash
   python -m venv face_detection_env

   # On Windows
   face_detection_env\Scripts\activate

   # On macOS/Linux
   source face_detection_env/bin/activate
   ```

3. **Install required packages**

   ```bash
   ```

```
pip install -r requirements.txt
```

Usage 💻

1. **Run the application**

   ```bash
   python main.py
   ```

2. **Choose from the menu**
   - **Option 1**: Detect faces in an image file
     - Enter the path to your image
     - View the result with detected faces highlighted
   - **Option 2**: Real-time face detection
     - Your webcam will start
     - Press 'q' to quit
     - Press 's' to save a screenshot

Project Structure 📁

```
face-detection-project/
│
├── main.py              # Main application file
├── requirements.txt     # Python dependencies
├── README.md            # Project documentation
├── output/              # Processed images (created automatically)
├── screenshots/         # Webcam screenshots (created automatically)
└── sample_images/       # Sample images for testing (optional)
```

How It Works 🔍

The project uses OpenCV's Haar Cascade classifiers for face detection:

1. **Haar Cascades**: Pre-trained classifiers that detect objects (faces/eyes)
2. **Image Processing**: Converts images to grayscale for better detection
3. **Detection Algorithm**: Scans the image at multiple scales
4. **Bounding Boxes**: Draws rectangles around detected faces
5. **Real-time Processing**: Processes video frames continuously

Key Functions 🛠️

- `detect_faces_in_image()`: Process static images
- `detect_faces_webcam()`: Real-time webcam detection
- Automatic creation of output directories
- Timestamp-based file naming
- Error handling for various scenarios

Sample Output 📸

The application will:

- Draw **blue rectangles** around detected faces
- Draw **green rectangles** around detected eyes
- Display the number of faces found
- Save processed images with timestamps

Troubleshooting 🔧

Webcam not working?

- Ensure your camera is not being used by another application
- Check camera permissions in your system settings

No faces detected?

- Ensure good lighting conditions
- Make sure faces are clearly visible and front-facing
- Try with different images

Installation issues?

- Make sure you have Python 3.7+
- Try upgrading pip: `pip install --upgrade pip`
- On some systems, you might need: `pip install opencv-python-headless`

Contributing 🤝

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/amazing-feature`)
3. Commit your changes (`git commit -m 'Add amazing feature'`)
4. Push to the branch (`git push origin feature/amazing-feature`)

5. Open a Pull Request

Future Enhancements 🚀

□ Face recognition (identify specific people)
□ Multiple face detection algorithms
□ GUI interface using tkinter
□ Batch processing for multiple images
□ Face mask detection
□ Age and gender prediction

License 📄

This project is open source and available under the MIT License.

Author 🧑‍💻

Created by [Your Name] - feel free to contact me!

Acknowledgments 🙏

- OpenCV community for the amazing computer vision library
- Haar Cascade classifiers for face detection
- Python community for excellent documentation

⭐ If you found this project helpful, please give it a star on GitHub!