Project Report: Face Detection API

1. Overview

The Face Detection API project is designed to identify human faces in uploaded images using computer vision techniques. The system is built with FastAPI as the backend and integrates with a Streamlit-based frontend for easy testing and demonstration.

---

2. Objective

- Detect faces in images
- Return the number of faces and their coordinates
- Prevent duplicate image processing using image hashing
- Provide a simple user interface for uploading images and viewing results

---

3. Technologies Used

- **FastAPI** for backend API
- **OpenCV** for face detection
- **Python Pillow (PIL)** for image handling
- **Hashlib** for duplicate image detection
- **Streamlit** for frontend testing
- **NumPy** for image array conversion

---

4. Key Features

✓ Face Detection

- Uses Haar Cascade classifier to detect faces in images
- Returns bounding box coordinates for each detected face

✓ Duplicate Image Detection

- Generates MD5 hash for each uploaded image
- Identifies if the same image is uploaded again
- Prevents reprocessing and returns a meaningful message

✓ Frontend Integration

- Streamlit UI allows users to upload images

- Calls the FastAPI backend for processing
- Displays the detection results in real time

---

5. Workflow

1. User uploads an image through Streamlit
2. The image is sent to the FastAPI endpoint `/detect-face`
3. API checks whether the image is new or already uploaded
4. If new:
    - Converts the image to grayscale
    - Runs face detection
    - Returns the number of faces and face coordinates
5. If duplicate:
    - Returns “Image already uploaded”

---

6. Output Format

The API returns:

```
{ "message": "New image processed", "faces_detected": <number>, "face_boxes": [[x, y, w, h], ...]
}
```

or, if duplicate:

```
{ "message": "Image already uploaded", "faces_detected": null, "face_boxes": null }
```

---

7. Conclusion

The Face Detection API project successfully demonstrates backend processing, duplicate detection logic, and frontend integration. It forms a solid foundation for further enhancements such as:

- Face recognition
- Image storage
- Drawing bounding boxes
- Deployment to cloud platforms