Name ID Kirubel Sentayehu GSR/7879/17 Surafel Sentayehu GSR/5928/17

Lab 12: Facial Recognition & Landmark Detection Report

What We Did

We implemented facial recognition and landmark detection using OpenCV DNN and Dlib. This lab covered the pipeline from face detection to real-time recognition systems.

Technologies Used

Technology	Purpose	Advantages
OpenCV DNN	Face Detection	Fast, accurate, built-in models
Dlib	Landmark Detection	Precise 68-point facial landmarks
face_recognition	Face Encoding	Simplified API, robust embeddings

What We Implemented

1. Face Detection with OpenCV

```
detector = cv2.CascadeClassifier('MobileNetSSD_deploy.prototxt',
'MobileNetSSD_deploy.caffemodel')
faces = detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
```

- Uses MobileNetSSD for fast face detection
- Draws green bounding boxes around detected faces
- Processes multiple image formats

2. Facial Landmark Detection

```
landmarks_list = face_recognition.face_landmarks(rgb_image, face_locations)
```

- Detects 68 facial landmark points
- Identifies key facial features:
  - Chin (17 points)
  - Eyebrows (10 points each)
  - Eyes (12 points each)
  - Nose (18 points)
  - Lips (20 points)
- Color-coded visualization for different features

3. Face Recognition System

```
encodings = face_recognition.face_encodings(image)
matches = face_recognition.compare_faces(known_encodings, face_encoding)
```

- Creates 128-dimensional face embeddings
- Compares unknown faces with known database
- Confidence scoring based on face distances
- Supports multiple people recognition

4. Real-Time Processing

```
cap = cv2.VideoCapture(0)
face_locations = face_recognition.face_locations(rgb_frame)
```

- Live webcam face recognition
- Optimized processing (every other frame)
- Real-time performance monitoring
- Interactive quit functionality

Test Results

Face Detection Results

- Successfully detected faces in all test images
- Robust performance across different lighting conditions
- Minimal false positives with optimized parameters

Landmark Detection Results

- Accurate 68-point landmark detection
- Consistent performance across different face orientations
- Clear visualization of facial structure

Recognition Results

- High accuracy for known faces in database
- Proper "Unknown" classification for unregistered faces
- Confidence scores provide reliability metrics

Insights

Observations

- Frame skipping for real-time processing
- Image resizing for faster computation
- Efficient encoding comparison algorithms

Observations II

- Multiple detection methods for robustness
- Confidence thresholding for reliability
- Landmark-based face alignment

Challenges and Solutions

Observations III

- **Problem**: High computational cost for face recognition
- **Solution**: Process every other frame and resize images

Observations IV

- **Problem**: Managing multiple faces in single image
- **Solution**: Iterate through all detected faces with individual processing

Observations V

- **Problem**: Distinguishing unknown faces from known ones
- **Solution**: Distance-based confidence scoring with thresholds

Conclusion

Implemented a complete facial recognition system with:

- face detection using OpenCV DNN
- landmark detection with Dlib
- face recognition using embeddings
- real-time processing capabilities

Exercise Completion Status

- **Exercise 1**: Multi-face recognition system implemented
- **Exercise 2**: Face encoding database created and tested
- **Exercise 3**: Real-time landmark display functional
- **Exercise 4**: Expression analysis framework established