



SUPERIOR UNIVERSITY

NAME :Muhammad ilyas

Roll number:033

Section:AI(4A)

SUBJECT:PAI-Lab

SUBMITTED TO :Mr. Rasikh Ali

Face Count Project Using OpenCV and Dlib

Introduction

This project is a **Face Counting System** that detects and counts the number of faces in a live webcam feed. It uses **OpenCV** for image processing and **Dlib** for face detection.

Libraries Used

1. **cv2 (OpenCV)** - Used for capturing video and processing images.
 2. **numpy** - Supports numerical operations (though not used in this code).
 3. **dlib** - Provides face detection functionality.
-

How It Works

1. The webcam is opened using **cv2.VideoCapture(0)**.
 2. The **dlib face detector** is initialized.
 3. The program continuously reads frames from the webcam.
 4. Each frame is converted to grayscale for better face detection.
 5. The **face detector** scans the image and finds faces.
 6. If faces are detected, a green box is drawn around each face, and a label is added showing the face count.
 7. The number of detected faces is displayed on the frame.
 8. The user can exit the program by pressing the **'q' key**.
-

Code Explanation

1. Importing Required Libraries

```
import cv2
import numpy as np
import dlib
```

- cv2 is used for handling images and videos.
- numpy is imported but not used.
- dlib is used for face detection.

2. Initializing Webcam and Face Detector

```
cap = cv2.VideoCapture(0)
detector = dlib.get_frontal_face_detector()
```

- `cv2.VideoCapture(0)`: Opens the webcam (0 for the default webcam).
- `dlib.get_frontal_face_detector()`: Loads the pre-trained face detection model.

3. Checking If Webcam Opened Successfully

```
if not cap.isOpened():
    print("Error: Could not open webcam.")
    exit()
```

- If the webcam is not working, the program prints an error message and stops.

4. Capturing Video Frames in a Loop

```
while True:
    ret, frame = cap.read()
```

- The program reads video frames continuously.
- `ret` is **True** if the frame is captured successfully.

5. Flipping and Converting the Frame to Grayscale

```
frame = cv2.flip(frame, 1)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

- `cv2.flip(frame, 1)`: Flips the image horizontally (mirror effect).
- `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`: Converts the frame to grayscale to improve face detection accuracy.

6. Detecting Faces

```
faces = detector(gray)
i = 0
```

- The detector scans the grayscale frame and detects faces.
- `faces` stores the detected face locations.
- `i` is used to count the number of faces.

7. Drawing Rectangles Around Faces

```
for face in faces:
```

```
x, y = face.left(), face.top()
x1, y1 = face.right(), face.bottom()
cv2.rectangle(frame, (x, y), (x1, y1), (0, 255, 0), 2)
```

- Loops through each detected face.
- Gets the **coordinates** (x, y) of the face.
- Draws a **green rectangle** around each face using cv2.rectangle().

8. Displaying the Face Count

```
i = i + 1
cv2.putText(frame, 'face num' + str(i), (x-10, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
print(face, i)
```

- Increases the face count (i for each face detected).
- Displays the face number on the frame using cv2.putText().

9. Displaying the Video Frame with Detected Faces

```
cv2.imshow('frame', frame)
```

- Shows the processed frame in a window.

10. Exiting the Program

```
if cv2.waitKey(15) & 0xFF == ord('q'):
    break
```

- If the user presses 'q', the loop breaks, and the program stops.

11. Releasing Resources

```
cap.release()
cv2.destroyAllWindows()
```

- cap.release(): Stops the webcam.
- cv2.destroyAllWindows(): Closes the display window.

How to Run the Program

1. Install dependencies (if not already installed):

```
pip install opencv-python dlib numpy
```

- 2.
 3. Run the script using:


```
python face.py
```
 - 4.
 5. A webcam window will open showing detected faces with a **count number**.
 6. Press '**q**' to exit the program.
-

Summary

- This project captures live video from the webcam.
 - It detects and counts the number of faces.
 - Uses **Dlib** for face detection and **OpenCV** for video processing.
 - Press '**q**' to stop the program.
-