Enhanced Face Unlock System with Object Recognition

By AMMASH BINTI ishfrat
Submitted on 22 July 2025

This project extends the basic face unlock system by integrating real-time object recognition using AI.

After successful face identification, the system activates the camera to detect and recognize objects in the frame.

It uses the 'face_recognition' and 'OpenCV' libraries for facial recognition and the 'MobileNet SSD' model for object detection.

1. Setup and Imports

```
import cv2
import face_recognition
import numpy as np
import time
```

2. Load Known Face Encodings

```
# Load a sample picture and learn how to recognize it.
known_image = face_recognition.load_image_file("known_user.jpg")
known_encoding = face_recognition.face_encodings(known_image)[0]
```

3. Initialize Object Detection Model

```
# Load pre-trained MobileNet SSD model and configuration
net              =              cv2.dnn.readNetFromCaffe("MobileNetSSD_deploy.prototxt",
"MobileNetSSD_deploy.caffemodel")

CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
           "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
           "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
           "sofa", "train", "tvmonitor"]
```

4. Start Camera and Perform Face Recognition

```
cap = cv2.VideoCapture(0)
print("Starting camera for face recognition...")

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

```
    rgb_frame = frame[:, :, ::-1]
    face_locations = face_recognition.face_locations(rgb_frame)
    face_encodings = face_recognition.face_encodings(rgb_frame, face_locations)

    for face_encoding in face_encodings:
        matches = face_recognition.compare_faces([known_encoding], face_encoding)
        if True in matches:
            authenticated = True
            print("Face recognized! Unlock successful.")
            break
    if authenticated:
        break
```

5. Perform Object Detection After Unlock

```
print("Starting object detection...")
while True:
    ret, frame = cap.read()
    (h, w) = frame.shape[:2]
     blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 0.007843, (300, 300),
127.5)
    net.setInput(blob)
    detections = net.forward()

    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > 0.5:
            idx = int(detections[0, 0, i, 1])
            label = CLASSES[idx]
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")
            cv2.rectangle(frame, (startX, startY), (endX, endY), (0, 255, 0), 2)
            y = startY - 15 if startY - 15 > 15 else startY + 15
             cv2.putText(frame, label, (startX, y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,
255, 255), 2)
            print(f"Detected: {label}")

    cv2.imshow("Object Recognition", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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