VEHICLE THEFT DETECTOR

Code for implementation:

```
import cv2
import smtplib
import face_recognition
import os
from email import encoders
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email.mime.text import MIMEText
from geopy.geocoders import Nominatim

fromaddr = "ecsproject2023@gmail.com"  # From Email ID
toaddr = "vpraneethnadh@gmail.com"  # To Email ID
filename = "/home/pi/ecs_images/captured_image.jpg"  # Update the file path here
password = "fthwnahozzuooxmc"  # Email Password
authorized_image_path = "/home/pi/ecs_images/4.jpg"  # Update the authorized image path here

def create_folders():
    folder = '/home/pi/ecs_images'
    if not os.path.exists(folder):
        os.makedirs(folder)
        print(f"Created folder: {folder}")

def sendEmail():
    try:
        print("Sending Email...")
        msg = MIMEMultipart()
        msg['From'] = fromaddr
```

```
        msg['To'] = toaddr
        msg['Subject'] = "Unauthorized Access Detected"

        body = "Unauthorized user detected. Please find the attached image and live location for reference."

        msg.attach(MIMEText(body, 'plain'))

        attachment = open(filename, "rb")
        p = MIMEBase('application', 'octet-stream')
        p.set_payload(attachment.read())
        encoders.encode_base64(p)
        p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
        msg.attach(p)

        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(fromaddr, password)
        text = msg.as_string()
        server.sendmail(fromaddr, toaddr, text)
        server.quit()
        print("Email Sent")

    except Exception as e:
        print("Email Sending Failed:", e)

def capture():
    print("Capturing Photo")
    cam = cv2.VideoCapture(0)
    ret_val, img = cam.read()

    # Make sure the directory '/home/pi/ecs_images' exists before capturing
    if not os.path.exists("/home/pi/ecs_images"):
        os.makedirs("/home/pi/ecs_images")
```

```
    cv2.imwrite(filename, img)
    cv2.destroyAllWindows()

def calculate_similarity(image1_path, image2_path):
    # Load the images
    image1 = face_recognition.load_image_file(image1_path)
    image2 = face_recognition.load_image_file(image2_path)

    # Encode the face in the images
    face_encodings1 = face_recognition.face_encodings(image1)
    face_encodings2 = face_recognition.face_encodings(image2)

    # Check if at least one face is detected in both images
    if not face_encodings1 or not face_encodings2:
        print("No face detected in one or both images.")
        return None

    # Use the first detected face (assuming single face in the images)
    face_encoding1 = face_encodings1[0]
    face_encoding2 = face_encodings2[0]
    # Calculate the Euclidean distance between the face encodings
    distance = face_recognition.face_distance([face_encoding1], face_encoding2)[0]

    # Calculate the similarity score as a percentage
    similarity_score = (1 - distance) * 100
    print(similarity_score)
    return similarity_score

def get_live_location():
    try:
        # Fetch the location details based on the IP address
        locator = Nominatim(user_agent="myGeocoder")
```

```
        location = locator.geocode("me")
        if location:
            location_info = {
                'latitude': location.latitude,
                'longitude': location.longitude,
                'city': location.raw['address'].get('city', ''),
                'state': location.raw['address'].get('state', ''),
                'country': location.raw['address'].get('country', '')
            }
            return location_info
        else:
            print("Failed to fetch the live location details.")
            return None

    except Exception as e:
        print("Error:", e)
        return None

if _name_ == "_main_":
    create_folders()
    capture()
    similarity_score = calculate_similarity(authorized_image_path, filename)
    live_location = get_live_location()
    if similarity_score is not None and live_location is not None:
        if similarity_score >= 70:
            print("Authorized User...Starting Engine...")
        else:
            print("Unauthorized User")
            sendEmail()

        print("Live Location Details:")
        print(f"Latitude: {live_location['latitude']}")
        print(f"Longitude: {live_location['longitude']}")
```

```
        print(f"City: {live_location['city']}")
        print(f"State: {live_location['state']}")
        print(f"Country: {live_location['country']}")
    else:
        print("Error: Unable to calculate similarity score or fetch live location.")
```

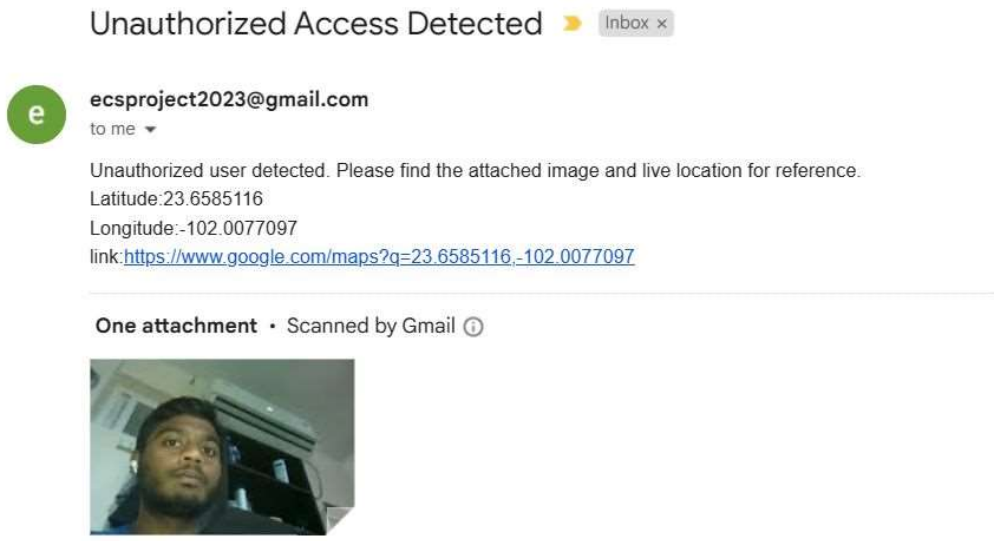
Images:

As it is an unauthorized user it shows the accuracy score and if it is less than 60 percent, it sends an email that unauthorized user found with an detected image.

```
>>> %Run ThiefDetector5.py

  Capturing Photo
  Photo captured succesfully...
  37.84089662874187
  Unauthorized User
  Sending Email...
  Email Sent
  Live Location Details:
  Latitude: 23.6585116
  Longitude: -102.0077097

>>>
```



As the captured image matches with the user, it accepts and doesn't send any mail.

```
>>> %Run ThiefDetector5.py

  Capturing Photo
  Photo captured succesfully...
  72.38989063551718
  Authorized User...Starting Engine...
  Live Location Details:
  Latitude: 23.6585116
  Longitude: -102.0077097
```

Teammates:

Vudattu Praneethnadh – 21BCE7762

Nathani Varshith – 21BCE7039

Shreyas Vanamala – 21BCE8510

Nishanth Reddy – 21BCE7079

Shyam Varma – 21BCE7073

G V Himaleswar Reddy – 21BCE9244