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']}")

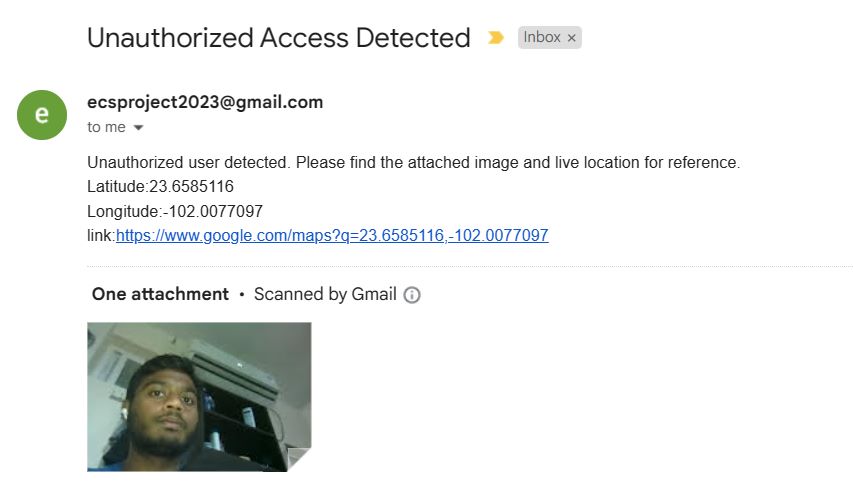
else:

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.

****

****

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

****

**Teammates:**

Vudattu Praneethnadh – 21BCE7762

Nathani Varshith – 21BCE7039

Shreyas Vanamala – 21BCE8510

Nishanth Reddy – 21BCE7079

Shyam Varma – 21BCE7073

G V Himaleswar Reddy – 21BCE9244