import cv2

# Motion detection cap = cv2.VideoCapture(0) fgbg = cv2.createBackgroundSubtractorMOG2() while True: ret, frame = cap.read() fgmask = fgbg.apply(frame) threshold = 1000 motion\_pixels = cv2.countNonZero(fgmask) if motion\_pixels > threshold: print("Motion Detected - Trigger Alert!") cv2.imshow('Frame', frame) cv2.waitKey(1)

cap.release()

cv2.destroyAllWindows()

# Capture video from camera camera\_source = 'http://ip\_camera\_url/video' cap = cv2.VideoCapture(camera\_source) if not cap.isOpened():

print("Error: Could not open camera.") exit() while True:

ret, frame = cap.read() if not ret:

print("Error: Could not read frame.") break

cv2.imshow('Camera Feed', frame) if cv2.waitKey(1) & 0xFF == ord('q'): break

cap.release() cv2.destroyAllWindows() # Face detection import dlib

detector = dlib.get\_frontal\_face\_detector()

predictor = dlib.shape\_predictor("shape\_predictor\_68\_face\_landmarks.dat")

face\_recognizer =

dlib.face\_recognition\_model\_v1("dlib\_face\_recognition\_resnet\_model\_v1.dat")

cap = cv2.VideoCapture("your\_video.mp4") while cap.isOpened(): ret, frame = cap.read() if not ret: break

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) faces = detector(gray) for face in faces: shape = predictor(frame, face)

face\_descriptor = face\_recognizer.compute\_face\_descriptor(frame, shape)

x, y, w, h = face.left(), face.top(), face.width(), face.height() cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2) cv2.imshow("Face Detection", frame)

if cv2.waitKey(1) & 0xFF == 27: # Press 'Esc' to exit break

cap.release()

cv2.destroyAllWindows()

# Licence plate detection import pytesseract

cap = cv2.VideoCapture("your\_video.mp4") pytesseract.pytesseract.tesseract\_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe' while cap.isOpened(): ret, frame = cap.read() if not ret: break

text = pytesseract.image\_to\_string(frame)

cv2.putText(frame, text, (50, 50), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2) cv2.imshow("License Plate Recognition", frame)

if cv2.waitKey(1) & 0xFF == 27: # Press 'Esc' to exit break cap.release()

cv2.destroyAllWindows()

# Email alert import smtplib def send\_email\_notification(subject, message): sender\_email = "email@gmail.com" sender\_password = "password" recipient\_email = "recipient\_email@gmail.com"

server = smtplib.SMTP("smtp.gmail.com", 587)

server.starttls()

server.login(sender\_email, sender\_password) email\_message = f"Subject: {subject}\n\n{message}" server.sendmail(sender\_email, recipient\_email, email\_message) server.quit()

cap = cv2.VideoCapture("your\_video.mp4") motion\_threshold = 1000 while cap.isOpened(): ret, frame = cap.read() if not ret: break

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) if previous\_frame is not None: frame\_diff = cv2.absdiff(previous\_frame, gray) motion\_value = cv2.countNonZero(frame\_diff) if motion\_value > motion\_threshold:

subject = "Motion Detected"

message = "Motion has been detected in the video stream." send\_email\_notification(subject, message) previous\_frame = ggra cv2.imshow("Event Detection", frame) if cv2.waitKey(1) & 0xFF == 27: break

cap.release() cv2.destroyAllWindows()

**TEST CASES IN OUR PROJECT:**

# 1. Motion detection

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| **(Motion pixels less than threshold pixels)**  Video frames | Should not show any alert. | No alert. |
| **(Motion pixels equals to threshold pixels)**  Video frames | Should show alert and display the frame. | Printed Motion Detected - Trigger Alert and displayed the frame. |
| **(Motion pixels equals to threshold pixels)**  Video frames | Should show alert and display the frame. | Printed Motion Detected - Trigger Alert and displayed the frame. |
| **(Pressing Esc button to exit)**  Esc | Should exit the program. | Exit the program. |

# 2. Capture video from camera

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| **(Incorrect ip\_camera\_url)** ip\_camera\_url | Should not connect to camera and should show errors. | Error: Could not open camera and  Error: Could not read frames. |
| **(Correct ip\_camera\_url)** ip\_camera\_url | Should connect to camera and should show captured frames. | Showing captured video frames. |
| **Pressing Esc button to exit)**  Esc | Should exit the program. | Exit the program. |

# 3. Face detection

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| **(No appearance of face in video frames)** Video frames. | Should not show any frame. | No frame. |
| **(Appearance of face in video frames)** Video frames. | Should show the frame in which face is detected. | Frame in which face detected. |
| **(Pressing Esc button to exit)**  Esc | Should exit the program. | Exit the program. |

# 4. Licence plate detection

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| **(No appearance of licence plate in**  **video frames)** Video frames. | Should not show any frame. | No frame. |
| **(Appearance of licence plate in video frames)**  Video frames. | Should show the frame in which licence plate is  detected and licence plate text. | Frame in which licence plate is detected with licence plate text. |
| **(Pressing Esc button to exit)** Esc. | Should exit the program. | Exit the program. |

# 5. Email message on motion detection

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| **(Motion pixels less than threshold pixels)**  Video frames | Should not show any alert and should not send email. | No alert and no email sent. |
| **(Motion pixels equals to threshold pixels)**  Video frames | Should show alert, display the frame and send email. | Printed Motion Detected -  Trigger Alert, displayed the frame and sent email. |
| **(Motion pixels equals to threshold pixels)**  Video frames | Should show alert, display the frame and send email. | Printed Motion Detected -  Trigger Alert, displayed the frame and sent email. |
| **(Pressing Esc button to exit)**  Esc | Should exit the program. | Exit the program. |