Bharati Vidyapeeth (Deemed to be University)

Department of Engineering and Technology

Kharghar, Navi Mumbai

**Department of Artificial Intelligence and Machine Learning**



**EXPERIMENT 10**

|  |  |
| --- | --- |
| **Name: Shivam Bhosle** | **Roll No: 02** |
| **Subject: COMPUTER VISION** | **Class/Batch: B1** |
| **Date of Performance:** | **Date of Submission:** |

|  |
| --- |
| **AIM** |

To perform Motion Detection in Python.

|  |
| --- |
| **Theory/Procedure/Algorithm** |

Motion detection using OpenCV is a process of detecting and analyzing changes in the position of objects in a video stream. OpenCV is a popular open-source computer vision library that provides various functions and tools for motion detection. The process of motion detection opencv involves capturing video frames, analyzing them for changes in pixel intensity, and detecting movement in the scene. Motion detection using OpenCV has numerous practical applications, such as surveillance systems, traffic monitoring, and video analysis. It can be implemented using different programming languages, including Python and C++.

**Steps For Motion Detection using OpenCV**

1. **Capture video frames:** The first step is to capture the video frames using OpenCV's video capture function.
2. **Pre-processing:** The next step is to pre-process the frames to prepare them for motion detection. This can include techniques such as resizing, grayscale conversion, and smoothing.
3. **Background modeling:** The next step is to model the background of the scene to detect moving objects. This can be done using techniques such as background subtraction or frame differencing.
4. **Thresholding:** The next step is to apply a threshold to the difference image obtained from the previous step. This helps to filter out small changes and highlight significant movements.
5. **Contour detection:** The next step is to detect contours in the thresholded image using OpenCV's contour detection function.

**CODE:**

import imutils

import cv2

import numpy as np

import time

FRAMES\_TO\_PERSIST = 10

MIN\_SIZE\_FOR\_MOVEMENT = 200

MOVEMENT\_DETECTED\_PERSISTENCE = 100

source = "C:\\Users\\Shivam 007\\Downloads\\video.avi"

cap = cv2.VideoCapture(source)

# Init frame variables

first\_frame = None

next\_frame = None

# Init display font and timeout counters

font = cv2.FONT\_HERSHEY\_SIMPLEX

delay\_counter = 0

movement\_persistent\_counter = 0

fourcc = cv2.VideoWriter\_fourcc('M','J','P','G')

fourcc = cv2.VideoWriter\_fourcc(\*'H264') # .mp4

fourcc = cv2.VideoWriter\_fourcc(\*'XVID') # .avi

out = None

while True:

transient\_movement\_flag = False

ret, frame = cap.read()

text = "Unoccupied"

if not ret:

print("CAPTURE ERROR")

continue

original\_frame = frame.copy()

frame = imutils.resize(frame, width = 750)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

gray = cv2.GaussianBlur(gray, (21, 21), 0)

if first\_frame is None: first\_frame = gray

delay\_counter += 1

if delay\_counter > FRAMES\_TO\_PERSIST:

delay\_counter = 0

first\_frame = next\_frame

next\_frame = gray

frame\_delta = cv2.absdiff(first\_frame, next\_frame)

thresh = cv2.threshold(frame\_delta, 25, 255, cv2.THRESH\_BINARY)[1]

thresh = cv2.dilate(thresh, None, iterations = 2)

cnts, \_ = cv2.findContours(thresh.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

for c in cnts:

(x, y, w, h) = cv2.boundingRect(c)

if cv2.contourArea(c) > MIN\_SIZE\_FOR\_MOVEMENT:

transient\_movement\_flag = True

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

if transient\_movement\_flag == True:

movement\_persistent\_flag = True

movement\_persistent\_counter = MOVEMENT\_DETECTED\_PERSISTENCE

if movement\_persistent\_counter > 0:

text = "Movement Detected " + str(movement\_persistent\_counter)

movement\_persistent\_counter -= 1

else:

text = "No Movement Detected"

cv2.putText(frame, str(text), (10,35), font, 0.75, (255,255,255), 2, cv2.LINE\_AA)

frame\_delta = cv2.cvtColor(frame\_delta, cv2.COLOR\_GRAY2BGR)

cv2.imshow("frame", frame)

if movement\_persistent\_counter>0 and not out:

height, width, \_ = frame.shape

out = cv2.VideoWriter(f'./recordings/{int(time.time())}\_video.avi',fourcc, 25.0 ,(width, height))

out.write(frame)

if movement\_persistent\_counter>0 and out:

out.write(frame)

ch = cv2.waitKey(1)

if ch & 0xFF == ord('q'):

if out is not None:

out.release()

out = None

break

cv2.waitKey(0)

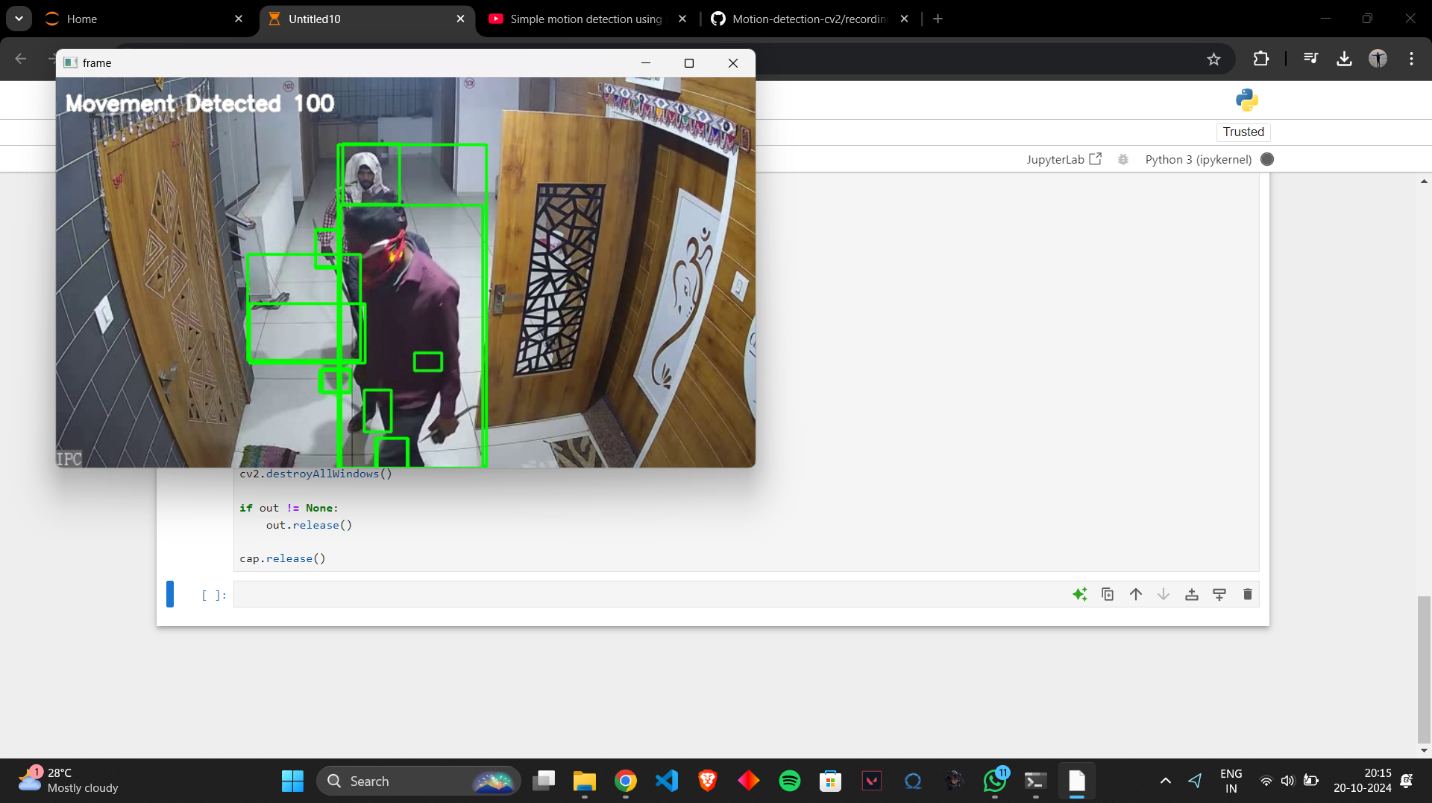
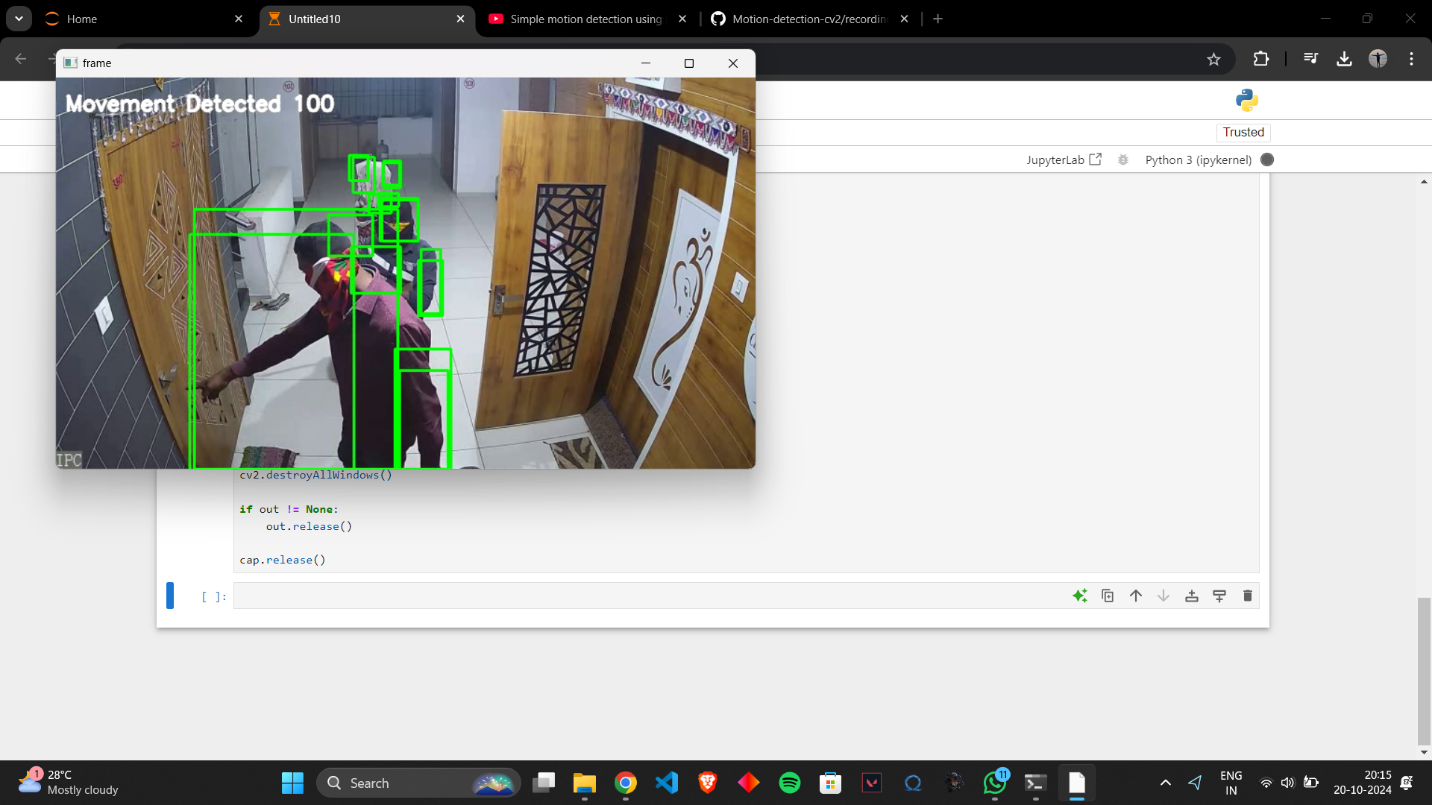
cv2.destroyAllWindows()

if out != None:

out.release()

cap.release()

**OUTPUT:**



|  |
| --- |
| **Conclusion** |

The experiment successfully demonstrated how to perform Motion Detection in Python.

|  |
| --- |
| **Assessment** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Timely Submission**  **(7)** | **Presentation**  **(06)** | **Understanding**  **(12)** | **Total**  **(25)** | **Sign** |
|  |  |  |  |  |