CGC ASSIGNMENT – 1

Name: RAHUL VARMA

Roll No: S20200010212

Motion Detection Assignment:

Motion detection is a technique used to detect changes in the position of an object or group of objects over time. It is typically used in security systems, surveillance cameras, and other applications where it is important to detect the presence or movement of objects in a scene.

There are several methods for motion detection, including:

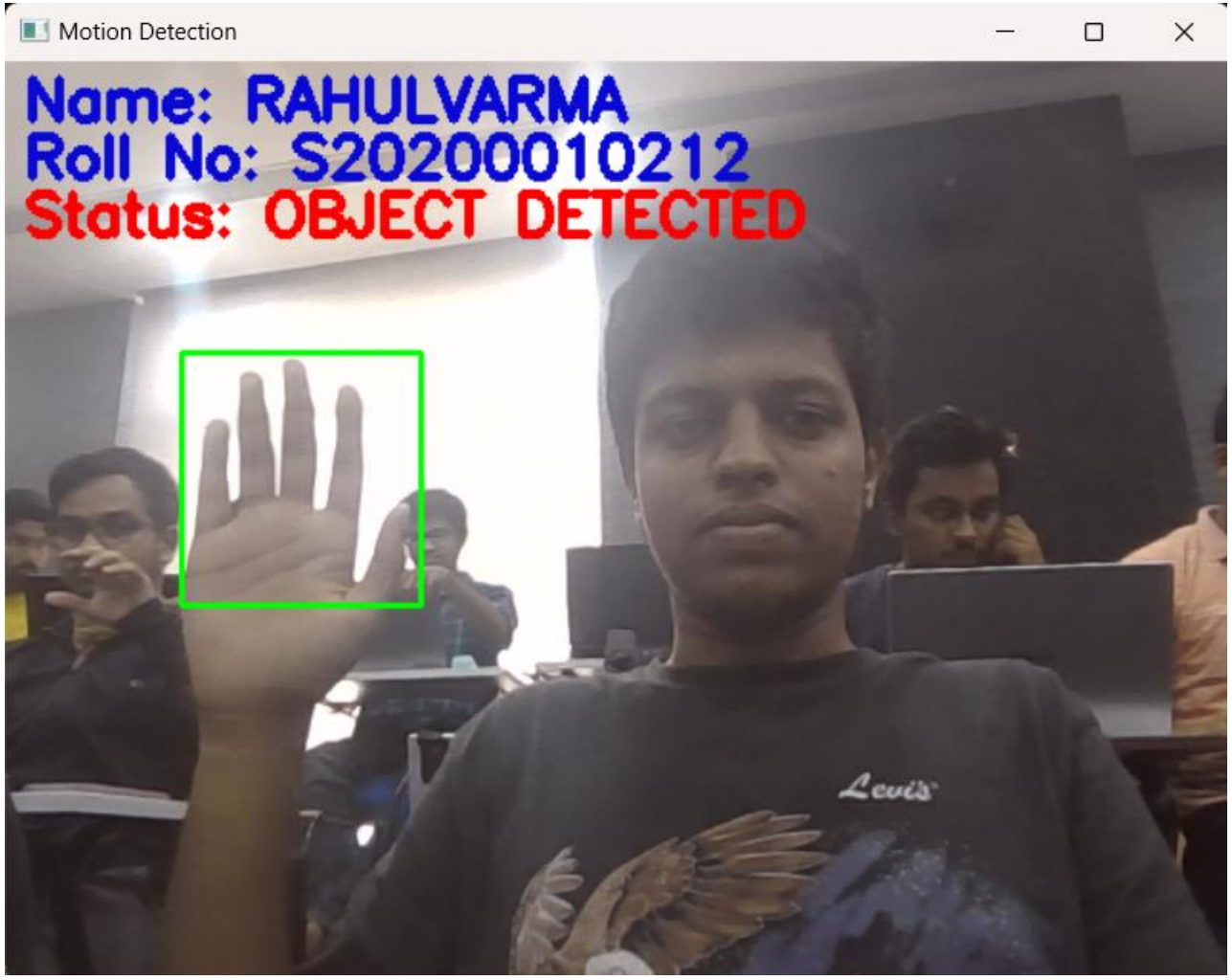
- Frame differencing: This method involves comparing consecutive frames of a video and identifying areas where there are significant differences between the two frames. These areas are likely to correspond to moving objects.

- Background subtraction: This method involves creating a static background model and subtracting it from each frame to identify areas of the frame that contain moving objects.
- Optical flow: This method involves tracking the movement of individual pixels within a frame to detect overall motion in the scene.
- Blob detection: This method involves identifying connected regions of pixels that have similar properties (such as color or intensity) and tracking their movement over time.

All these methods have their own advantages and disadvantages and the choice of method depends on the specific application and the type of data.

Threshold is changed accordingly for only the object detection which is moving the frames from one second to another second.

Motion Detected:



Code:

```python
import cv2

# Initialize webcam
cap = cv2.VideoCapture(0)

while True:
    # Read first frame
    ret, frame1 = cap.read()
    gray1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)

    # Read next frame
    ret, frame2 = cap.read()
    gray2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)

    # Calculate difference between frames
    diff = cv2.absdiff(gray1, gray2)

    # Threshold the difference to identify motion
    thresh = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)[1]

    # Dilate the thresholded image to fill in holes
    dilated = cv2.dilate(thresh, None, iterations=2)

    # Find contours in the dilated image
    cnts, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Draw contours on the original frame
    for c in cnts:
        if cv2.contourArea(c) < 2500:
            continue
        (x, y, w, h) = cv2.boundingRect(c)
        cv2.rectangle(frame1, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(frame1, "Status: {}".format('OBJECT DETECTED'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
    cv2.putText(frame1, "Name: {}".format('RAHULVARMA'), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (215, 10, 10), 3)
    cv2.putText(frame1, "Roll No: {}".format('S20200010212'), (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (215, 10, 10), 3)

    # Show the frame
    cv2.imshow("Motion Detection", frame1)

    # Update the previous frame
    gray1 = gray2

    # Exit if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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

Code:

```
import cv2

# Initialize webcam
cap = cv2.VideoCapture(0)

while True:
    # Read first frame
    ret, frame1 = cap.read()
    gray1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)

    # Read next frame
    ret, frame2 = cap.read()
    gray2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)

    # Calculate difference between frames
    diff = cv2.absdiff(gray1, gray2)

    # Threshold the difference to identify motion
    thresh = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)[1]

    # Dilate the thresholded image to fill in holes
    dilated = cv2.dilate(thresh, None, iterations=2)

    # Find contours in the dilated image
    cnts, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Draw contours on the original frame
    for c in cnts:
        if cv2.contourArea(c) < 2500:
            continue
```

```
        (x, y, w, h) = cv2.boundingRect(c)
        cv2.rectangle(frame1, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(frame1, "Status: {}".format('OBJECT DETECTED'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
0, 255), 3)
    cv2.putText(frame1, "Name: {}".format('RAHULVARMA'), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (215, 10, 10),
3)
    cv2.putText(frame1, "Roll No: {}".format('S20200010212'), (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (215, 10,
10), 3)

    # Show the frame
    cv2.imshow("Motion Detection", frame1)

    # Update the previous frame
    gray1 = gray2

    # Exit if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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