PRACTICAL NO: 9

AIM: Intelligent Video Surveillance (Real time activity and behavior recognition)

----------CODE----------

```
import cv2

def intelligent_video_surveillance(video_source=0):
    cap = cv2.VideoCapture(video_source)

    if not cap.isOpened():
        print("Error: Could not open video source.")
        return

    ret, frame1 = cap.read()
    if not ret:
        print("Error: Could not read frame from video source.")
        cap.release()
        return

    frame1_gray = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
    frame1_gray = cv2.GaussianBlur(frame1_gray, (21, 21), 0)

    while cap.isOpened():
        ret, frame2 = cap.read()
        if not ret:
            print("Warning: Empty frame, ending surveillance.")
            break

        frame2_gray = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
        frame2_gray = cv2.GaussianBlur(frame2_gray, (21, 21), 0)

        frame_diff = cv2.absdiff(frame1_gray, frame2_gray)

        _, thresh = cv2.threshold(frame_diff, 25, 255, cv2.THRESH_BINARY)
        thresh = cv2.dilate(thresh, None, iterations=2)

        contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

        for contour in contours:
            if cv2.contourArea(contour) < 500:
                continue
            (x, y, w, h) = cv2.boundingRect(contour)
            cv2.rectangle(frame2, (x, y), (x + w, y + h), (0, 255, 0), 2)

        cv2.imshow("Intelligent Video Surveillance", frame2)

        frame1_gray = frame2_gray

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

intelligent_video_surveillance()
```

----------OUTPUT----------