SOURCE PYTHON CODE:

```
import cv2
import serial
33
# Load the pre-trained Haar Cascade classifiers for human detection
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
fullbody_cascade = cv2.CascadeClassifier('haarcascade_fullbody.xml')
lowerbody_cascade = cv2.CascadeClassifier('haarcascade_lowerbody.xml')
horizontal_cascade = cv2.CascadeClassifier('haarcascade_horizontal.xml')
# Open a video capture device (use 0 for the default camera)
cap = cv2.VideoCapture(0)
# Initialize the human count to 0
human_count = 0
# Open a serial connection to the Arduino
ser = serial.Serial('COM3', 9600) # Replace COM3 with the name of your serial
port
while True:
 # Read a frame from the video stream
 ret, frame = cap.read()


 # Prepare the frame for object detection
 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

 # Run object detection on the frame for human face
 face_detections = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)

 # Run object detection on the frame for full body
 fullbody_detections = fullbody_cascade.detectMultiScale(gray,
scaleFactor=1.1, minNeighbors=5, minSize=(30, 30),
```

```
flags=cv2.CASCADE_SCALE_IMAGE)

 # Run object detection on the frame for lower body
 lowerbody_detections = lowerbody_cascade.detectMultiScale(gray,
scaleFactor=1.1, minNeighbors=5, minSize=(30, 30),
flags=cv2.CASCADE_SCALE_IMAGE)

 # Run object detection on the frame for horizontal human horizontal_detections
= horizontal_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)


 # Post-process the detections to filter out non-human objects
 human_count = 0
 for (x, y, w, h) in face_detections:
 cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
 human_count += 1

 for (x, y, w, h) in fullbody_detections:
 cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
 human_count += 1

 for (x, y, w, h) in lowerbody_detections:
 cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
 human_count += 1

 for (x, y, w, h) in horizontal_detections:
 cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
 human_count += 1
 # Display the resulting frame with the human count
```

```
 cv2.putText(frame, f"Human Count: {human_count}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
 cv2.imshow("Human Detection", frame)
 # Exit the program if the "q" key is pressed
 if cv2.waitKey(1) & 0xFF == ord('q'):
 break
cap.release()
cv2.destroyAllWindows()
ser.close()
```

Arduino.ino:

```
void setup(){
 pinMode(12,OUTPUT); // RELAY PIN
 Serial.begin(9600);
}
void loop(){
 digitalWrite(12,HIGH);
 delay(3000);
 int human_count;
 if (Serial.available() > 0) {
 human_count = Serial.parseInt();

}
 if(human_count >= 0){
 digitalWrite(12,LOW);
 delay(2000);
 digitalWrite(12,HIGH);
}
}
```