```
# you can set custom kernel size if you want
kernel = None

# initilize background subtractor object
foog = cv2.createBackgroundSubtractorMOG2(
    detectShadows=True, varThreshold= 50, history=200)

# Noise filter threshold
# thresh = 1100
thresh = 1100

while(1):
    ret, frame = cap.read()
    if not ret:
        break

    dim = (width, height)
    frame = cv2.resize(frame, dim, interpolation=cv2.INTER_AREA)

    # Apply background subtraction
    fgmask = foog.apply(frame)

    # Get rid of the shadows
    ret, fgmask = cv2.threshold(fgmask, 250, 255, cv2.THRESH_BINARY)

    # Apply some morphological operations to make sure you have a good mask
    # fgmask = cv2.erode(fgmask,kernel,iterations = 1)
    fgmask = cv2.dilate(fgmask, kernel, iterations=4)

    # Detect contours in the frame
    contours, hierarchy = cv2.findContours(
        fgmask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    if contours:

        # Get the maximum contour
        cnt = max(contours, key=cv2.contourArea)
        # print(cnt)
        # make sure the contour area is somewhat hihger than some threshold to
make sure its a person and not some noise.
        if cv2.contourArea(cnt) > thresh:
```