

# THÀNH VIÊN NHÓM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MSSV | HỌ TÊN | EMAIL | ĐIỆN THOẠI | CHỮ KÍ |
| 14520401 | Trương Ngọc Kha | [14520401@gm.uit.edu.vn](mailto:14520401@gm.uit.edu.vn) |  |  |
| 14520422 | Huỳnh Đăng Khoa | [14520422@gm.uit.edu.vn](mailto:1312447@student.fit.hcmus.edu.vn) |  |  |

1. **Mã nguồn :**

Link download : <https://drive.google.com/drive/folders/1CPizAFkhdwcS5tabpeDbCT7b3HVKnb7E?usp=sharing>

1. **Nội dung:**

**Đề tài :** đếm số người người di chuyển lên xuống, ra vào qua một vùng nhất định.

1. **Mở và xữ lý trên video hoặc camera của máy**

* Sử dụng class VideoCapture của thư viện OpenCV theo như ở [đây](https://docs.opencv.org/3.1.0/d8/dfe/classcv_1_1VideoCapture.html#aeb1644641842e6b104f244f049648f94) để mở video file bằng câu lệnh cv2.VideoCapture(‘test.avi’) hoặc cv2.VideoCapture(0) để sử dụng trực tiếp camera của máy.
* Lấy các thông tin như độ phân giải của video để phục vụ cho việc phân các vùng xữ lý khi có vật di chuyển qua vùng đó. Vẽ các vùng phù hợp mà các đối tượng (ở đây là con người) thường di chuyển qua để phục vụ cho việc đếm.
* Khi video đang được mở, quét từng frame một để lấy thông tin của video cho đến hết video.

**import** **numpy** **as** **np**

**import** **cv2**

cap = cv2.VideoCapture(test.avi') #mở video file

**while**(cap.isOpened()):

ret, frame = cap.read() #đọc từng frame

**try**:

cv2.imshow('Frame',frame)

cv2.imshow('Background Substraction',fgmask)

**except**:

#hết video

**print**(‘end’)

**break**

#dừng chương trình bằng nút Q hoặc ESC

k = cv2.waitKey(**30**) & **0xff**

**if** k == **27**:

**break**

cap.release() #đóng video file

cv2.destroyAllWindows() #đóng tất cả các cửa sổ

1. **Loại bỏ background**

* Tài liệu tham khảo cho phần này : <https://docs.opencv.org/3.1.0/db/d5c/tutorial_py_bg_subtraction.html#gsc.tab=0>
* Để xác định được con người theo đề bài, chúng ta cần phải loại bỏ background. Background là các khung cảnh tĩnh và thường không chuyển động nhiều. Các vật chuyển động nhiều như con người sẽ được xữ lý để đếm

**import** **numpy** **as** **np**

**import** **cv2**

cap = cv2.VideoCapture(test.avi') #mở video file

fgbg = cv2.createBackgroundSubtractorMOG2(detectShadows = True) #tạo loại bỏ background

**while**(cap.isOpened()):

ret, frame = cap.read() #đọc từng frame

fgmask = fgbg.apply(frame) #sử dụng loại bỏ background

**try**:

cv2.imshow('Background Substraction',fgmask)

**except**:

#hết video

**print**(‘end’)

**Break**

cv2.imshow('Frame',frame)

#dừng chương trình bằng nút Q hoặc ESC

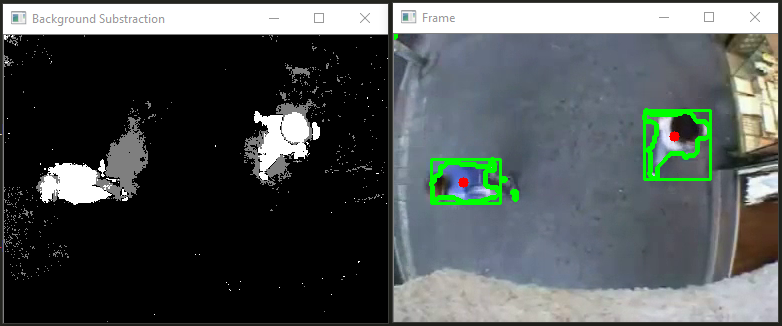
k = cv2.waitKey(**30**) & **0xff**

**if** k == **27**:

**break**

cap.release() #đóng video file

cv2.destroyAllWindows() #đóng tất cả các cửa sổ



1. **Khử noise và làm mịn**

* Tài liệu tham khảo cho phần này : <https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html>  
  <https://docs.opencv.org/2.4/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html>
* Chúng ta sử dụng Morphological Transformations để khử các noise như các chấm trắng và bóng của các vật di chuyển như hình trên. Khử noise có ích rất nhiều trong việc xác định vật và con người
* Sử dụng ngưỡng phù hợp để cho ra kết quả vừa ý nhất. Sử dụng cv2.THRESH\_BINARY để phân rõ ra các vùng trắng và đen (tham khảo thêm ở [đây](https://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html))

**import** **numpy** **as** **np**

**import** **cv2**

cap = cv2.VideoCapture(test.avi') #mở video file

#khởi tạo các kernel cho morphographic filters

kernelOp = np.ones((**3**,**3**),np.uint8)

kernelCl = np.ones((**11**,**11**),np.uint8)

fgbg = cv2.createBackgroundSubtractorMOG2(detectShadows = True) #tạo loại bỏ background

**while**(cap.isOpened()):

ret, frame = cap.read() #đọc từng frame

fgmask = fgbg.apply(frame) #sử dụng loại bỏ background

**try**: ret,imBin=cv2.threshold(fgmask,190,255,cv2.THRESH\_BINARY)

mask1 = cv2.morphologyEx(imBin, cv2.MORPH\_OPEN, kernelOp)

mask = cv2.morphologyEx(mask1 , cv2.MORPH\_CLOSE, kernelCl)

cv2.imshow('Background Substraction',fgmask)

**except**: #hết video

**print**(‘end’)

**break**

cv2.imshow('Frame',frame)

cv2.imshow('Mask',mask)

cv2.imshow('Mask1',mask1)

#dừng chương trình bằng nút Q hoặc ESC

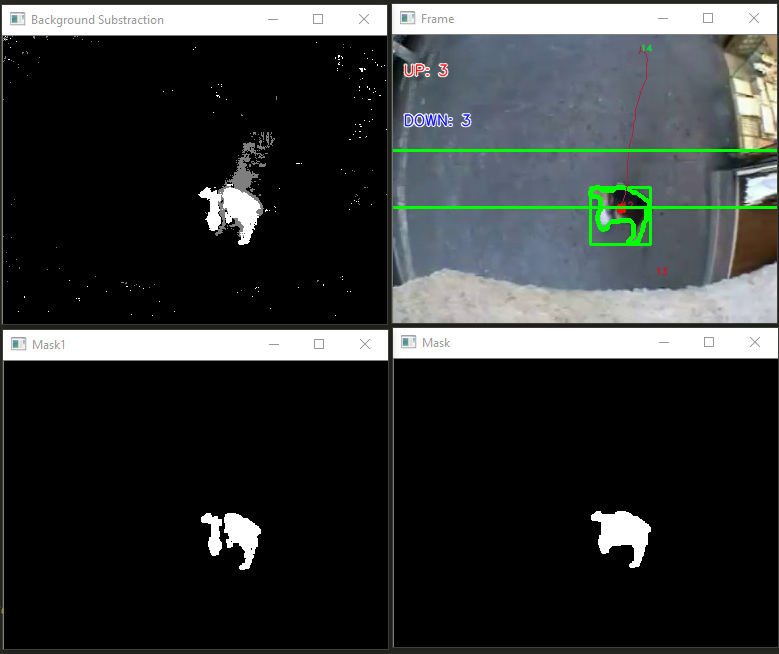
k = cv2.waitKey(**30**) & **0xff**

**if** k == **27**:

**break**

cap.release() #đóng video file

cv2.destroyAllWindows() #đóng tất cả các cửa sổ



1. **Vẽ các đường viền quanh các vật chuyển động**

* Đến đây thì chúng ta đã loại bỏ được background, nhiễu và chỉ còn lại các chuyển động.
* Bây giờ, ta cần vẽ đường viền quanh các vật chuyển động bằng hàm cv2.findContours(), cv2.drawContours() của thư viện OpenCV (xem thêm về phần Contuors của OpenCV ở [đây](https://docs.opencv.org/3.1.0/d4/d73/tutorial_py_contours_begin.html)).

\_, contours0, hierarchy = cv2.findContours(mask,cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_NONE)

**for** cnt **in** contours0:

cv2.drawContours(frame, cnt, -**1**, (**0**,**255**,**0**), **3**, **8**)

1. **Xác định con người**

* Tiếp theo, ta cần xác định contour nào là người. Ở đây ta sẽ cho một kích thước tối thiểu (ngưỡng tối thiểu), nếu contour nào lớn hơn ngưỡng này thì xem như là người (phương pháp này khá đơn giản nhưng lại có thể dễ dàng dẫn đến sai sót khi các vật chuyển động không phải người có kích thước lớn hơn ngưỡng đã quy định).
* Sau đó ta sẽ vẽ một hình chữ nhật bao quanh các contour nào được xác định là người.
* Mỗi video có chất lượng và độ phân giải khác nhau và vị trí camera cũng khác nhau nên cần phải xác định một ngưỡng phù hợp để cho ra kết quả tốt nhất. Không quá lớn cũng không quá nhỏ. Ta có thể tùy ý thay đổi ngưỡng theo từng video khác nhau để cho kết quả tốt nhất.
* Cụ thể, ở đây ta sẽ lấy ngưỡng là:



Với h là chiều cao của video, w là chiều rộng video.

\_, contours0, hierarchy = cv2.findContours(mask,cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_NONE)

**for** cnt **in** contours0:

cv2.drawContours(frame, cnt, -**1**, (**0**,**255**,**0**), **3**, **8**)

area = cv2.contourArea(cnt)

**print** area

**if** area > areaTH:

M = cv2.moments(cnt)

cx = int(M['m10']/M['m00'])

cy = int(M['m01']/M['m00'])

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

cv2.circle(frame,(cx,cy), **5**, (**0**,**0**,**255**), -**1**)

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

cv2.imshow('Frame',frame)

,**255**,**0**), **3**, **8**)

1. **Theo dõi chuyển động**

* Phần này chúng ta cần theo dõi chuyển động của người đó để sau đó có thể xác định người đó đi lên hay xuống, vào hay ra. Cụ thể, ta sẽ lưu trữ danh sách các vị trí của một người khi người đó di chuyển.
* Ở frame đầu tiên khi ta xác định được một người nào đó, ta cần cho người đó một ID và lưu trữ vị trí ban đầu của người đó.
* Ở những frame tiếp theo, để theo dấu người đó, ta cần so khớp contour của người đó với ID đã lưu trữ trước đó, đồng thời cặp nhật vị trí của người đó. Danh sách các vị trí đã được lưu trữ của một người sau này dùng để xác định người người đó đi lên hay xuống, vào hay ra.
* Cụ thể, ở những frame tiếp theo này, ta so khớp các contour đã được xác định là người với những người đã được lưu trữ ở frame trước. Nếu không khớp với bất kỳ người nào ở fame trước thì đây là người mới, và ta cần tạo một ID mới.
* Class MyPerson trong file Person.py dùng để lưu trữ ID và vị trí của một người.

1. **Đếm số lượng người di chuyển qua một khu vực**

* Để xét xem một người di chuyển lên hay xuống, ta sẽ tạo ra hai đường ngang(ta có thể thay đổi hướng và vị trí của 2 đường này tùy vào hướng di chuyển của các đối tượng trong video).
* Khi một người di chuyển, ta xem người này chạm vượt qua đường nào trước (dựa vào danh sách các vị trí đã được lưu). Nếu một đối tượng vượt qua đường nằm ở dưới trước thì đối tượng đó đang đi lên, ngược lại, nếu đối tượng vượt qua đường nằm ở trên trước thì đối tượng đang đi xuống.
* Khi một người vượt qua 1 đường( nằm trên hoặc ở dưới) thì ta đã đếm được một người đi xuống/lên.

1. **Hướng phát triển trong tương lai**

Như đã đề cặp trong phần 5 của chương II, cách xác định con người bằng ngưỡng tối thiểu gây ra nhiều hạn chế như trong trường hợp các vật chuyển động không phải người có kích thước lớn hơn ngưỡng đã quy định hoặc vật chuyển động là con người nhưng lại có kích thước nhỏ hơn ngưỡng. Do đó, để xác định một cách chính xác hơn các vật chuyển động là con người thì ta cần xem xét đến các đặc điểm khác của con người và thậm chí, nâng cao hơn, ta có thể sử dụng machine learning để nhận dạng các đặc điểm này.