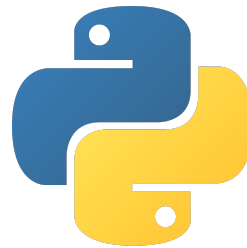

Reconocimiento de peatones y comportamiento anómalo

EDISON OMAR LOZA GUZMÁN



Índice

1. Código	3
2. Manual de Uso	4

1. Código

Algunas de las funciones a destacar son las siguientes:

```

1 def background_subtraction(previous_frame, frame_resized_grayscale, min_area):
2     frameDelta = cv2.absdiff(previous_frame, frame_resized_grayscale)
3     thresh = cv2.threshold(frameDelta, 25, 255, cv2.THRESH_BINARY)[1]
4     thresh = cv2.dilate(thresh, np.ones((11,11)), 5)
5     _, cnts, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
6     temp = 0
7     for c in cnts:
8         if cv2.contourArea(c) > 500:
9             temp = 1
10    return temp

```

Con ayuda de esta función y mediante la diferencia entre los dos frames se puede saber si se ha realizado un movimiento. Uno de los aspectos a tener en cuenta de esta función es la calibración de los parámetros de las funciones usadas, tales como, **cv2.threshold** donde los parametros (25, 255) pueden ser sustituidos por otros según las necesidades del ambiente para despreciar ciertos tonos de color.

Finalmente en **if cv2.contourArea(c) > 500** se puede ajustar un "threshold" a partir del cual no ignorar los contornos de las figuras encontradas al producirse un movimiento. Un valor aproximado puede ser 500

```

1 def detect_people(frame):
2     (rects, weights) = hog.detectMultiScale(frame, winStride = (4, 4), padding = (0, 0),
3         scale = 1.05)
4     rects = np.array([[x, y, x + w, y + h] for (x, y, w, h) in rects])
5     pick = non_max_suppression(rects, probs = None, overlapThresh = 0.65)
6     return pick

```

Con ayuda de un detector de objetos *HOG (Histogram of Oriented Gradients)* podemos detectar una persona en el frame donde se ha producido un movimiento. Para ello debemos declarar un objeto de tipo HOG como se hace en la línea 3, **hog.detectMultiScale(...)**, donde se puede ajustar los parámetros dependiendo de la condiciones de trabajo. En el siguiente enlace se explica detalladamente los parámetros de la función y los distintos valores que pueden adoptar según las necesidades. **Parámetros de HOG.**

```

1 def drawDetection(frame, xA, yA, xB, yB, cont):
2     # Dibujamos en la imagen.
3     os.environ['TZ'] = 'EST'
4     strTime = " "
5     strTime = time.strftime("%x %X")
6     cv2.putText(frame, "Time: {time}".format(time=tiempoDiff), topLeftCornerOfText1, cv2.
7         FONT_HERSHEY_SIMPLEX, fontScale, color, grosorNormal) #Peligro
8     # Guardamos la imagen en un folder.
9     cv2.imwrite("frame%d.png" %cont, frame)
10    return frame

```

Una vez que se detecta a la persona se muestra la hora en la que se produce la captura de pantalla, y se guarda la imagen de forma local. Más adelante dicha imagen deberá ser enviada al servidor de imágenes. El nombre de las imágenes a guardar son frame1.png frame2.png ... frameN.png.

```
1 def moveX(xA, yA, pick, frame):
2
3
4     # Ver el punto actual supera la linea
5     x = 0
6     if xA > xS1:
7         x = 1
8
9     # Ver que el punto anterior no se mueve del sitio, aqui tengo devuelvo un 0 si esta
        de paso
10    # Variacion en el eje de la X
11    variacionX = 5
12    if abs(xA - coordenadasAnteriores[0]) > variacionX:
13        #cv2.putText(frame, "XXX", (xA, yA), cv2.FONT_HERSHEY_SIMPLEX, fontScale, color,
            grosorNormal) #Peligro
14        x = 2
15
16    return x
```

Tanto en el eje X como en el Y, la comprobación de una situación anómalo es idéntico, se comprueba que el punto anterior donde se encontraba la persona en el frame anterior esta ligeramente cerca del nuevo punto en el frame actual. Esta comprobación se realiza a partir de que el sujeto se encuentre durante más de un cierto tiempo dentro de una *"zona de peligro"*. Estos dos parámetros y otros muchos viene definidos dentro del código.

2. Manual de Uso

Primeramente cabe destacar que las versiones usada son: **Python 3.6.7** y **3.2.0** Para el uso del código se ejecuta el siguiente comando estando dentro de la carpeta donde se encuentra el programa.

```
1 odroid@odroid:~$ pwd
2 ~/opencv_detectarmovimientos/src
```

```
1 odroid@odroid:~/opencv_detectarmovimientos/src$ python CapturaMovimientoHOG.py
```

NOTA: Se puede cerrar la ventana de vídeo pulsando la tecla "S".