

Workshop Détection de mouvement

OpenCV



I – Requirements

- Python3
- pip3
- `pip3 install opencv-python`
- `pip3 install imutils`
- avoir `live_feed.mp4` dans son dossier

II – Code

```
import cv2
import imutils

# Prends un input vidéo
video = cv2.VideoCapture("./live_feed.mp4")

# Initialise la frame de référence
FirstFrame = None

while 42:
    # Read une frame de la vidéo
    ret, frame = video.read()
    # Si la vidéo arrive à la fin, exit
    if ret == False:
        exit()
    # Par défaut, set le display text à "Sleeping..."
    DisplayText = "Sleeping..."

    # Convertir la frame en grayscale, puis la flouter
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (25, 25), 100)

    # Initialiser la frame de référence si elle n'existe pas
    if FirstFrame is None:
        FirstFrame = gray
        continue

    # Calculer la différence entre la fframe de référence et la frame actuelle
    frameDelta = cv2.absdiff(FirstFrame, gray)
    # Accentuer les différences entre les 2 frames
    thresh = cv2.threshold(frameDelta, 25, 255, cv2.THRESH_BINARY)[1]

    # Trouver les contours sur la frame contenant les différences
    contours = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    contours = imutils.grab_contours(contours)

    # Si un contour est assez grand, changer le display text
    for c in contours:
        if cv2.contourArea(c) < 3000:
            continue
        DisplayText = "Recording!"

    # Display le texte sur la frame actuelle, puis display le tout
    cv2.putText(frame, DisplayText, (10, frame.shape[0] - 20),
cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0, 0, 255), 3)
    cv2.imshow("Big Brother", frame)

    # Check si la touche q est enfoncée puis quitter si c'est le cas
```

```
key = cv2.waitKey(1)
```

```
if key == ord('q'):
```

```
    exit()
```