import cv2, time, pandas

from datetime import datetime

first\_frame = None

status\_list = [None, None]

times = []

df = pandas.DataFrame(columns=["Start", "End"])

video = cv2.VideoCapture(0) #for reading video from the camera

while True:

check, frame = video.read()

status = 0

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

gray = cv2.GaussianBlur(gray,(21,21),0) #blurring the image to find the delta frame as it reduces the noise

if first\_frame is None:

first\_frame = gray

continue

delta\_frame = cv2.absdiff(first\_frame,gray) #creating the delta frame by finding the diff b/w blurry gray and first\_frame

thresh\_frame = cv2.threshold(delta\_frame, 30, 255, cv2.THRESH\_BINARY)[1] #changing the difference of first\_frame-current\_frame > 30 to white(255) and it acceps a tupple thats why [1]

thresh\_frame = cv2.dilate(thresh\_frame, None, iterations=2)

(\_,cnts,\_)=cv2.findContours(thresh\_frame.copy(),cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

for contour in cnts:

if cv2.contourArea(contour) < 10000:

continue

status = 1

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

cv2.rectangle(frame, (x, y), (x+w, y+h), (0,255,0), 3)

status\_list.append(status)

if status\_list[-1]==1 and status\_list[-2]==0:

times.append(datetime.now())

if status\_list[-1]==0 and status\_list[-2]==1:

times.append(datetime.now())

cv2.imshow("Gray Frame", gray)

cv2.imshow("Delta Frame", delta\_frame)

cv2.imshow("Threshold Frame", thresh\_frame)

cv2.imshow("Color Frame", frame)

key = cv2.waitKey(1)

if key == ord('q'): #will break when pressed q from the keyboard

if status == 1:

times.append(datetime.now())

break

print(status\_list)

print(times)

for i in range(0, len(times), 2):

df = df.append({"Start":times[i], "End":times[i+1]}, ignore\_index=True)

df.to\_csv("Times.csv")

video.release()

cv2.destroyAllWindows()