import cv2

import numpy as np

def nothing(x):

pass

cap=cv2.VideoCapture(0)

cap.set(3,720)

cap.set(4,480)

kernel=np.ones((5,5),np.uint8)

canvas=None

x1,y1=0,0

noiseth =800

cv2.namedWindow("Trackbars")

cv2.createTrackbar("L - H", "Trackbars", 0, 179, nothing)

cv2.createTrackbar("L - S", "Trackbars", 0, 255, nothing)

cv2.createTrackbar("L - V", "Trackbars",0,255,nothing)

cv2.createTrackbar("U - H", "Trackbars", 179, 179, nothing)

cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)

cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

while True:

ret, frame=cap.read()

if not ret:

break

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

l\_h = cv2.getTrackbarPos("L - H", "Trackbars")

l\_s = cv2.getTrackbarPos("L - S", "Trackbars")

l\_v = cv2.getTrackbarPos("L - V", "Trackbars")

u\_h = cv2.getTrackbarPos("U - H", "Trackbars")

u\_s = cv2.getTrackbarPos("U - S", "Trackbars")

u\_v = cv2.getTrackbarPos("U - V", "Trackbars")

if canvas is None:

canvas=np.zeros\_like(frame)

hsv=cv2.cvtColor(frame,cv2.COLOR\_BGR2HSV)

lower\_range = np.array([l\_h, l\_s, l\_v])

upper\_range = np.array([u\_h, u\_s, u\_v])

mask = cv2.inRange(hsv, lower\_range, upper\_range)

mask = cv2.erode(mask, kernel, iterations=1)

mask = cv2.dilate(mask, kernel, iterations=2)

contours, hierarchy = cv2.findContours(

mask, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

if contours and cv2.contourArea(max(contours,key = cv2.contourArea)) > noiseth:

c = max(contours, key=cv2.contourArea)

x2, y2, w, h = cv2.boundingRect(c)

if x1 == 0 and y1 == 0:

x1, y1 = x2, y2

else:

canvas = cv2.line(canvas, (x1, y1), (x2, y2), [0, 0, 255], 4)

x1, y1 = x2, y2

else:

x1, y1 = 0, 0

frame = cv2.add(frame, canvas)

#res = cv2.bitwise\_and(frame, frame, mask=mask)

#mask\_3 = cv2.cvtColor(mask, cv2.COLOR\_GRAY2BGR)

stacked = np.hstack((canvas, frame))

cv2.imshow('canvas',canvas)

cv2.imshow('frame', frame)

key=cv2.waitKey(1)

if key== 27:

break

if key==ord('c'):

canvas=None

cap.release()

cap.destroyAllWindows()