|  |
| --- |
| 基本資料 |
| * 姓名：王豪逸 * 標題：B024020014\_Assignment 11(基本) |
| 作業目的與設計 |
| * 目標：手部辨識+控制GPIO開關 * 電路設計說明(總分1分)：   LED燈GPIO: 綠21  視訊鏡頭+麥克風  喇叭   電路設計照片(總分1分)：   * 程式邏輯設計(總分1分)：   影像處理先減小圖片大小為原來得一半，再使用convertToGray, blurred, threshold簡化影像，找出凹凸多邊形，並取的他的質心。使用convexityDefects找出手指間的縫隙個數用來辨識手指，為了減少干擾多邊形缺陷要離質心夠近才會被歸類為縫隙，因此只能辨識只有手掌的畫面。圖片每秒擷取10張，每秒統計一次縫隙數量，手指數量為縫隙-1。2指打開LED、4指關閉LED。   * 影片URL：https://youtu.be/-kHX0fkCPas |
| 心得(總分1分) |
| 一開始簡化影像的參數沒有完全按照範例設置，常常手指有一半被辨識出來，按照範例社後就好多了。計算手指也常常把不必要的地方計算進去(手腕  )，簡化偵測範圍就好多了。 |
| 程式碼 |
| 程式碼：  ====基本題===(總分1分)  # -\*- coding: utf-8 -\*-  import cv2  import numpy as np  import time, os  import RPi.GPIO as GPIO  from gtts import gTTS  filename = "tmp.mp3"  led = 21  class WedCam():  def \_\_init\_\_(self):  self.cap = cv2.VideoCapture(0)  self.handSize = 80  def get\_frame(self):  sucess = False  while not sucess:  sucess, frame = self.cap.read()  frame = cv2.resize(frame, (0,0), fx=0.5, fy=0.5)  gray = cv2.cvtColor(frame,cv2.COLOR\_BGR2GRAY)  blur = cv2.GaussianBlur(gray, (37,37),0)  ret, binary = cv2.threshold(blur,120,255,cv2.THRESH\_BINARY\_INV+cv2.THRESH\_OTSU)  contours, hierarchy = cv2.findContours(binary, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  maxArea = 0  maxC = None  for c in contours:  area = cv2.contourArea(c)  if area > maxArea:  maxArea = area  maxC = c  if maxC is None:  return 0  hull = cv2.convexHull(maxC)  moments = cv2.moments(maxC)  if moments['m00'] != 0:  cx = int(moments["m10"]/moments['m00'])  cy = int(moments["m01"]/moments['m00'])  cent = (cx, cy)  cv2.circle(frame, cent, self.handSize, [255,0,255], 2)  maxC = cv2.approxPolyDP(maxC,0.01\*cv2.arcLength(maxC, True),True)  cv2.drawContours(frame,[maxC],-1,(0,0,255),3)  cv2.drawContours(frame,[hull],-1,(0,255,255),3)  hull = cv2.convexHull(maxC,returnPoints=False)  defects = cv2.convexityDefects(maxC, hull)  hole = 0  for de in defects:  s,e,f,d = de[0]  start = tuple(maxC[s][0])  end = tuple(maxC[e][0])  far = tuple(maxC[f][0])  # dist = cv2.pointPolygonTest(maxC, (cx, cy), True)  if ((cent[0] - far[0]) \*\* 2 + (cent[1] - far[1]) \*\* 2) < self.handSize \*\* 2:  hole += 1  cv2.circle(frame, far, 5, [255,255,255],-1)  cv2.imshow("img", frame)  return hole  def Say(str):  aud = gTTS(text=str, lang="zh-tw")  aud.save(filename)  os.system("mpg321 "+filename)  print(str)  def setLed(mode):  if mode == 3:  GPIO.output(led, True)  Say("兩隻手指頭，綠色LED打開了")  elif mode == 5:  GPIO.output(led, False)  Say("四隻手指頭，綠色LED關閉了")  if \_\_name\_\_ == '\_\_main\_\_':  w = WedCam()  GPIO.setmode(GPIO.BCM)  GPIO.setup(led, GPIO.OUT)  GPIO.output(led, False)  timecount = 0  handArray = []  while True:  timecount += 1  try:  handArray.append(w.get\_frame())  except Exception as e:  print(e)  if cv2.waitKey(1) == 27:  break    if timecount == 10:  hand = np.array(handArray).mean().round()  setLed(hand)  timecount = 0  handArray = []  time.sleep(0.1) |