高一理甲 電腦課期末專案

Python編程－會飛的蟒蛇



指導老師：梁智杰老師

報告組員：高曉鋒

一、設計理念

無人機市場持續成長，未來社會對無人機的依賴將會越來越大。現今主流無人機主要依賴人手操作。藉著本次報告我希望可以結合無人機和視覺化人工智能，以達到無人控制追蹤的目的，使無人機真正做到「無人」機。

二、程式簡介及目標對象

以wifi連接電腦與無人機。無人機返回圖像經由電腦端opencv python插件分析後返回人臉位置。經由算法判斷無人機指令後向無人機發送指令

三、程式功能介紹

main.py

| import detectionvol1 as detectionfile import cv2 import DroneControl as DroneControl import KeyboardInputHandler as KeyboardInputHandler from GuiHandler import main as Gui import keyboard from time import sleep from time import time |
| --- |

插入插件

| tools = Gui()  NoDrone = False if tools == "drone" else True |
| --- |

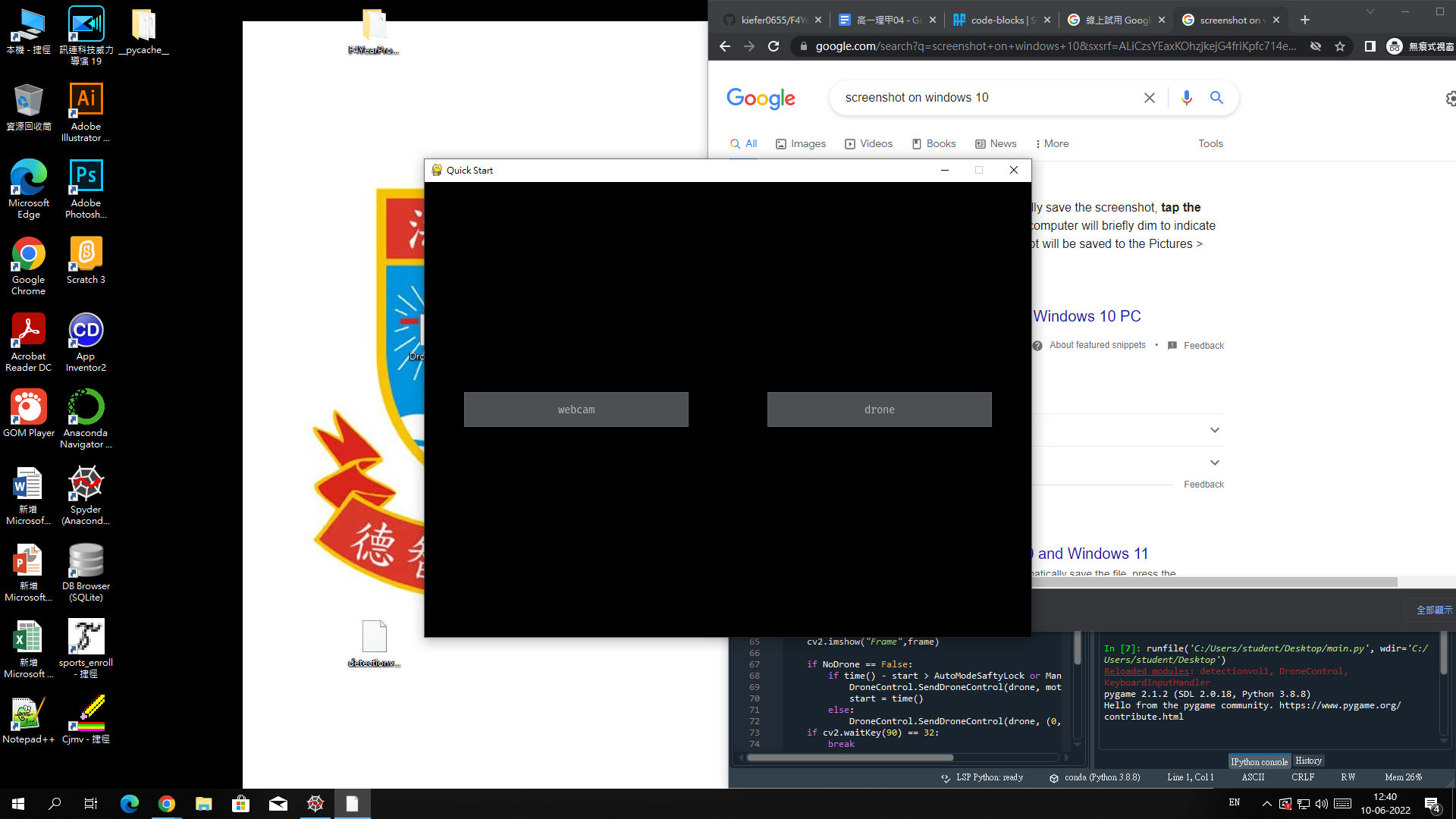
運行GUIHandler.py中使用Pygame插件製作的GUI

GUIHandler.py

| import pygame\_gui import pygame import sys  def main():  pygame.init()    pygame.display.set\_caption('Quick Start')  window\_surface = pygame.display.set\_mode((800, 600))    background = pygame.Surface((800, 600))  background.fill(pygame.Color('#000000'))    manager = pygame\_gui.UIManager((800, 600))    drone\_button = pygame\_gui.elements.UIButton(relative\_rect=pygame.Rect((450, 275), (300, 50)),  text='drone',  manager=manager)  webcam\_button = pygame\_gui.elements.UIButton(relative\_rect=pygame.Rect((50, 275), (300, 50)),  text='webcam',  manager=manager)    clock = pygame.time.Clock()  is\_running = True    pygame.init()    choice = ""    while is\_running:  time\_delta = clock.tick(60)/1000.0  for event in pygame.event.get():  if event.type == pygame.QUIT:  pygame.quit()  sys.exit(0)    if event.type == pygame\_gui.UI\_BUTTON\_PRESSED:  if event.ui\_element == drone\_button:  choice = "drone"  is\_running = False  if event.ui\_element == webcam\_button:  choice = "webcam"  is\_running = False      manager.process\_events(event)    manager.update(time\_delta)    window\_surface.blit(background, (0, 0))  manager.draw\_ui(window\_surface)    pygame.display.update()    pygame.quit()    return choice |
| --- |

製作視窗及按鈕，等待直至按下。按下後返回選擇，跳出迴圈。

運行結果



main.py

| NoDrone = False if tools == "drone" else True |
| --- |

決定是否啟動除錯模式

| AutoModeSaftyLock = 0.7  if NoDrone == False:  drone = DroneControl.GetDroneAndConnect()  print(drone.get\_battery()) else:  cap = cv2.VideoCapture(0)  def GetFrame():  if NoDrone == False:  frame = DroneControl.GetFrame(drone)  else:  ret, frame = cap.read()  return frame   print("Ready to be Take Off \nSpace to Take Off \nEscape to Escape") |
| --- |

根據是否為除錯模式決定連接方式

| while True:  frame = GetFrame()  cv2.imshow("Frame",frame)   if cv2.waitKey(90) == 32:  break  if cv2.waitKey(90) == 27:  if NoDrone == True:  cap.release()  cv2.destroyAllWindows()  exit() print("Take Off")  if NoDrone == False:  drone.takeoff() sleep(0.5) if NoDrone == False:  DroneControl.SendDroneControl(drone, (0,0,0,0)) sleep(0.5) ManualControl = True print("Control Onlined")  start = time() |
| --- |

連接無人機取得視頻信號後等待起飛

| while True:  frame = GetFrame()   if ManualControl == True:  motion = KeyboardInputHandler.GetMotionByKeyboard()   ManualControl = KeyboardInputHandler.ManualControlControler(ManualControl)   frame , motion = detectionfile.DetectAndDraw(frame,ManualControl,motion)   cv2.imshow("Frame",frame)   if NoDrone == False:  if time() - start > AutoModeSaftyLock or ManualControl == True:  DroneControl.SendDroneControl(drone, motion)  start = time()  else:  DroneControl.SendDroneControl(drone, (0,0,0,0))  if cv2.waitKey(90) == 32:  break   sleep(0.05)  if NoDrone == False:  DroneControl.Landing(drone) else:  cap.release() cv2.destroyAllWindows() |
| --- |

主迴圈

主迴圈拆解

| frame = GetFrame() |
| --- |

按照宣告方式取得影像

| if ManualControl == True:  motion = KeyboardInputHandler.GetMotionByKeyboard |
| --- |

如果正在使用手動模式，取得輸入

KeyboardInputHandler.py

| def GetMotionByKeyboard():  motion = [0,0,0,0]   LR, FB, UD, Yaw = 0, 0, 0, 0   if keyboard.is\_pressed('a'):  LR -= ManualMotionPower   if keyboard.is\_pressed('d'):  LR += ManualMotionPower   if keyboard.is\_pressed('w'):  FB += ManualMotionPower   if keyboard.is\_pressed('s'):  FB -= ManualMotionPower   if keyboard.is\_pressed('up'):  UD += ManualMotionPower   if keyboard.is\_pressed('down'):  UD -= ManualMotionPower   if keyboard.is\_pressed('left'):  Yaw -= ManualMotionPower   if keyboard.is\_pressed('right'):  Yaw += ManualMotionPower   motion = [LR, FB, UD, Yaw]  return motion |
| --- |

按鍵盤輸入取得並返回指令

mian.py

| ManualControl = KeyboardInputHandler.ManualControlControler(ManualControl) |
| --- |

KeyboardInputHandler.py

| def ManualControlControler(ManualControl):  if keyboard.is\_pressed('tab'):  ManualControl = not ManualControl  return ManualControl |
| --- |

如果按下TAB鍵轉換控制模式

main.pyfre , motion

| frame , motion = detectionfile.DetectAndDraw(frame,ManualControl,motion) |
| --- |

取得標記後的畫面以其AI運算後的無人機指令

detectionvol1.py

| def DetectAndDraw(frame,ManualControl,motion):   midpoint = (int(frame.shape[1]/2),int(frame.shape[0]/2))   cv2.circle(frame, midpoint, radius = 5, color = RED , thickness = -1)   frame, Detected, ItemMid = Detection(model = "face", frame = frame)   action = ["",""]   if Detected == True:  LineDeltaHigh = int(frame.shape[0]/3)  LineDeltaWeight = int(frame.shape[1]/3)   if ItemMid[0] < LineDeltaWeight:  action[1] = "Left"  if ItemMid[0] > 2 \* LineDeltaWeight:  action[1] = "Right"   if ItemMid[1] < LineDeltaHigh:  action[0] = "Up"  if ItemMid[1] > 2 \* LineDeltaHigh:  action[0] = "Low"   if ManualControl == False:  motion = GetMotion(ItemMid, midpoint, action)   command = action[0]+action[1]   frame = Draw(frame,ItemMid, midpoint, command , motion, ManualControl)   return frame , motion |
| --- |

DetectAndDraw 重點解釋

| frame, Detected, ItemMid = Detection(model = "face", frame = frame) |
| --- |

| def Detection(model,frame):  if model == "face":  gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5)   Detected = False   mid = None   for (x, y, w, h) in faces:   mid = (int(x + w/2),int(y + h/2))   cv2.rectangle(frame, (x,y), (x + w,y + h), GREEN , 2)   Detected = True   return frame, Detected, mid |
| --- |

取得面部位置並使用方框標記

| action = ["",""]   if Detected == True:  LineDeltaHigh = int(frame.shape[0]/3)  LineDeltaWeight = int(frame.shape[1]/3)   if ItemMid[0] < LineDeltaWeight:  action[1] = "Left"  if ItemMid[0] > 2 \* LineDeltaWeight:  action[1] = "Right"   if ItemMid[1] < LineDeltaHigh:  action[0] = "Up"  if ItemMid[1] > 2 \* LineDeltaHigh:  action[0] = "Low" |
| --- |

| command = action[0]+action[1] |
| --- |

取得除錯顯示使用對無人機的指令

| if ManualControl == False:  motion = GetMotion(ItemMid, midpoint, action) |
| --- |

| def GetMotion(ItemMid, midpoint, command):  motion = [0,0,0,0]  if command[1] != "":  if ItemMid[0] < midpoint[0]:  motion[0] = -MotionPower  if ItemMid[0] > midpoint[0]:  motion[0] = MotionPower  if command[0] != "":  if ItemMid[1] < midpoint[1]:  motion[2] = MotionPower  if ItemMid[1] > midpoint[1]:  motion[2] = -MotionPower  return motion |
| --- |

如果非手動控制，取得無人機指令

| frame = Draw(frame,ItemMid, midpoint, command , motion, ManualControl) |
| --- |

| def Draw(frame,ItemMid, mid, command, motion ,ManualControl):  LineDeltaHigh = int(frame.shape[0]/3)  LineDeltaWeight = int(frame.shape[1]/3)   cv2.line(frame, (LineDeltaWeight,0), (LineDeltaWeight,frame.shape[0]), BLUE, 2)  cv2.line(frame, (2 \* LineDeltaWeight,0), (2 \* LineDeltaWeight,frame.shape[0]), BLUE, 2)  cv2.line(frame, (0,LineDeltaHigh), (frame.shape[1],LineDeltaHigh), BLUE, 2)  cv2.line(frame, (0,2 \* LineDeltaHigh), (frame.shape[1],2 \* LineDeltaHigh), BLUE, 2)   if ItemMid != None:  cv2.circle(frame, mid, radius = 10, color = RED , thickness = -1)  cv2.line(frame, ItemMid, mid, RED, 5)   cv2.putText(frame, command , (50,70),cv2.FONT\_HERSHEY\_SIMPLEX, 2 , GREEN, 1, cv2.LINE\_AA)  cv2.putText(frame, str(motion) , (50,frame.shape[0] - 50),cv2.FONT\_HERSHEY\_SIMPLEX, 1 , RED, 2, cv2.LINE\_AA)   ManualControlText = "Manual Control: ON" if ManualControl else "Manual Control: OFF"   cv2.putText(frame, ManualControlText , (frame.shape[1] - 350,60),cv2.FONT\_HERSHEY\_SIMPLEX, 1 , RED, 1, cv2.LINE\_AA)   return frame |
| --- |

繪畫除錯畫面

main.py

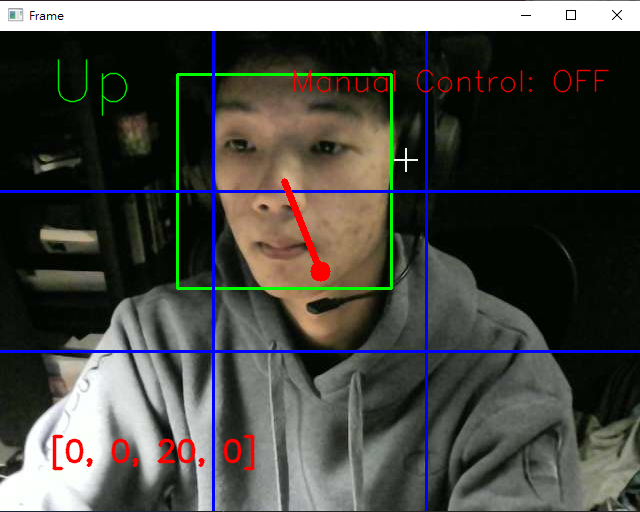
| cv2.imshow("Frame",frame)   if NoDrone == False:  if time() - start > AutoModeSaftyLock or ManualControl == True:  DroneControl.SendDroneControl(drone, motion)  start = time()  else:  DroneControl.SendDroneControl(drone, (0,0,0,0)) |
| --- |

顯示畫面以及向無人機發送指令

| if cv2.waitKey(90) == 32: |
| --- |

如果SPACEBAR被按下跳出迴圈

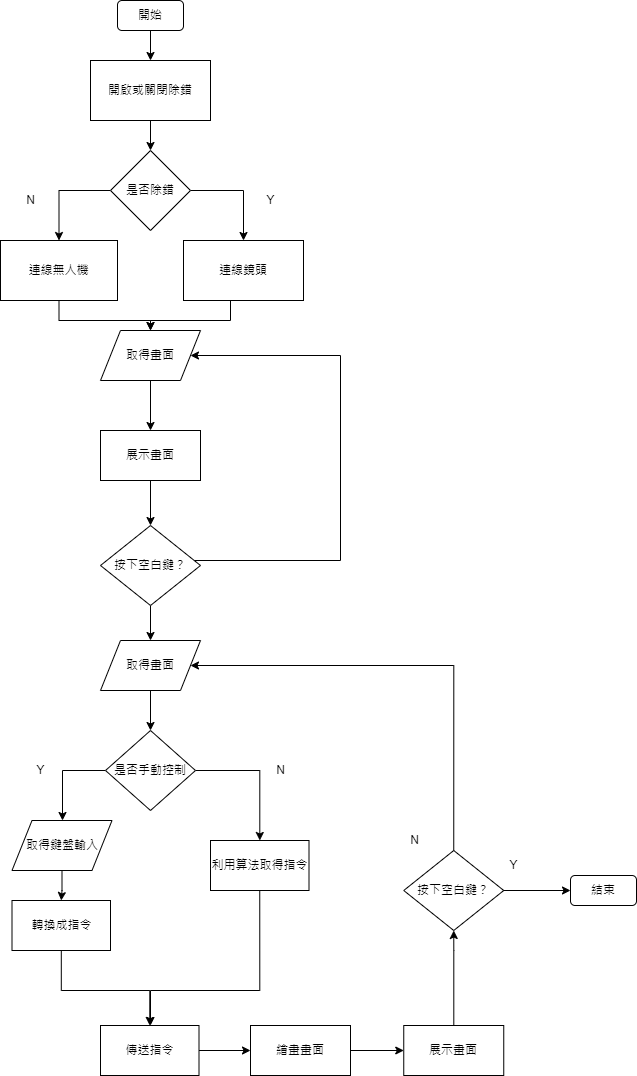
主迴圈執行結果



| if NoDrone == False:  DroneControl.Landing(drone) else:  cap.release() cv2.destroyAllWindows() |
| --- |

無人機降落，清理並關閉程式

四、流程圖



五、難點解釋

這是我第一次使用Github作版本管理以及備份，在初期使用時感到有點難度但在多次反覆使用後，我初步掌握了Github的使用方法。

這也是我第一次接觸Python語言驅動的IOT程式，當中出現了很多各式各樣的奇怪漏洞，經常需要多番覆查Github Repository和Stack Overflow 才能解決問題。

六、參考來源

Dji Tello Python Github Repository

<https://github.com/damiafuentes/DJITelloPy>

Tello Tutroial <https://www.youtube.com/watch?v=LmEcyQnfpDA&t=3422s>

CV2 Tutroial

<https://www.youtube.com/watch?v=oXlwWbU8l2o>