

無人機PYTHON編程及人工智能(AI)應用

當今社會無人機應用



更多...



目標和使命

面對時代的呼應

我們正在經歷一個偉大的時代，冰冷的機器人逐漸從工廠和荒野走進人們的生活，走進人們溫暖的家，成為人類生活的陪伴者。如何培養能制造和設計飛行器和機器人的工程師，這是一個比設計飛行器和機器人本身更大的問題。為了回答這個問題，大疆開推出了 RoboMasters 機器人比賽和STEAM項目，為頂尖高校的青年工程師和孩子們提供一個展現自我的舞台。

讓更多青年工程師成為明星

我們希望讓更多埋頭苦幹的青年工程師走進公眾視野，更多社會資源能夠向他們傾斜，獲得關注認可。系統性、大規模地培養具有真知灼見的人才，是當今我們教育和文化領域的一塊短板。

創始人的情懷

大疆創始人汪滔在港科大就讀之時就是機器人比賽的常客，也正是在機器人比賽中取得的研究成果讓他有了創業的想法並一舉成功，因此也催生了他想要讓更多人成為工程師並享受成功的可能。



無人機編程教育



高清 720P



500万有效像素



最长飞行 13 分钟



精准视觉悬停



编队飞行



挑战卡多样玩法



智能飞行模式



丰富的可扩展性



艾思能達(香港)有限公司
ASK Idea(Hong Kong)Limited

Tello EDU編程及AI應用展示



四軸無人機用途操控守則



根據民航處網

(https://www.cad.gov.hk/english/Unmanned_Aircraft_Systems.html)

- ✦ 重量：不超過七千克（不計燃料）
- ✦ 操作範圍：無人機系統一般不得在機場交通區域或任何機場5公里範圍內操作。
- ✦ 操作高度：無人機系統的放飛高度不得超過地面以上300呎（約90米）。
- ✦ 操作時間：無人機系統的操作時間只限白晝。



無人駕駛飛機系統操作安全提示

✓  **≤7公斤**



✓ 肉眼可看見無人機
(非第一身視角)

✓ 放飛前為無人機和電池作安全檢查

   100%

✓  適合放飛：
• 白晝
• 天氣良好

✓ 空曠地方
(遠離人多擠迫的地方)

 **≤300呎**
(~90米)
地面以上

安全放飛無人機

✗  不適合放飛：
• 晚間
• 天氣不佳
• 雲霧中

✓ 遠離人群、船隻、車輛或構築物

 50米 50米

✗ 不得在機場、飛機升降航道或直升機坪附近放飛



✓ 遵守通訊事務管理局及其他要求

   私隱

! 危害他人或財產安全，可被檢控

✎ 操作手冊 出租或受酬用途須有民航處許可證

如需更多資訊，請電郵 uas@cad.gov.hk 及瀏覽網頁：

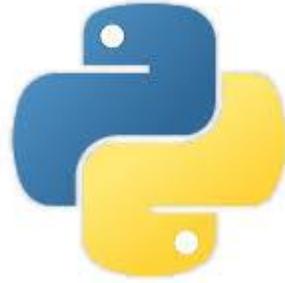
http://www.cad.gov.hk/chinese/Unmanned_Aircraft_Systems.html



課堂四軸無人機用途操控守則

- ▶ 未經老師許可，不可啟動無人機。
- ▶ 禁止不受控的飛行活動。
- ▶ 不可利用無人機碰撞到其他同學。
- ▶ 禁止向飛行中的無人機拋擲物品。
- ▶ 需遵照教師指示作飛行活動。

什麼是Python



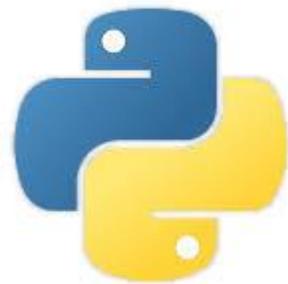
- ▶ Python 是一種廣泛使用的直譯式，開放通用型程式語言，第一版釋出於1991年。
- ▶ Python的設計強調程式碼的可讀性和簡潔的語法。相比於C++或Java，Python讓開發者能夠用更少的代碼表達想法。
- ▶ Python擁有能夠自動管理記憶體使用，並且支援多種編程範式，包括物件導向、命令式、函數式和程序式編程。其本身擁有一個巨大而廣泛的資料庫。

Python優點



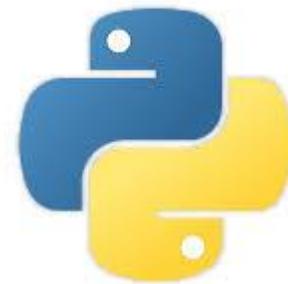
- ▶ 簡易編程。Python 具有許多物件導向的特性，然而並不要求一定得用物件導向的方式撰寫。撇開物件導向的議題不談，Python 通常可以靠比較短的程式碼完成比較多的功能，或者可以寫得比較清楚。
- ▶ 功能強大。Python 自 1990 年由 Guido van Rossum 在荷蘭的 CWI 開始發展以來，從 0.9 進步到今天的 2.4.2，不但累積了相當完整的標準程式庫 (模組)，更有無以計數的非標準模組，而且絕大部分都是開放原始碼的。單以內建的模組來講，從簡單的數學運算、字串處理、網際網路協定連線、網際網路資料處理、各種壓縮格式，以及 POSIX 與主要作業系統的支援功能等等，含括的範圍非常地廣泛。
- ▶ 跨平台。各種主要的作業系統都支援 Python。Python 程式常常不需要修改，便可以同時在 Linux 與 Windows 平台上執行，即使撰寫 GUI 程式 (透過 PyGTK, wxPython 等 binding) 也是一樣。所撰寫的 Python 程式透過標準的 distutils (模組) 進行包裝後，用標準的方式即可安裝於各種平台；在 Windows 下更可以自動產生方便的可執行 installer。
- ▶ 容易擴充。Python 算是執行效率不錯的直譯式語言，但畢竟比不上 C 和 Fortran。然而只要我們想，大可以 C/C++ 或 Fortran 撰寫高效率的模組；這些模組的使用方式，與內建模組以及用 Python 撰寫的模組完全一樣。最好的是，撰寫的方法並不困難。

Python編寫程式

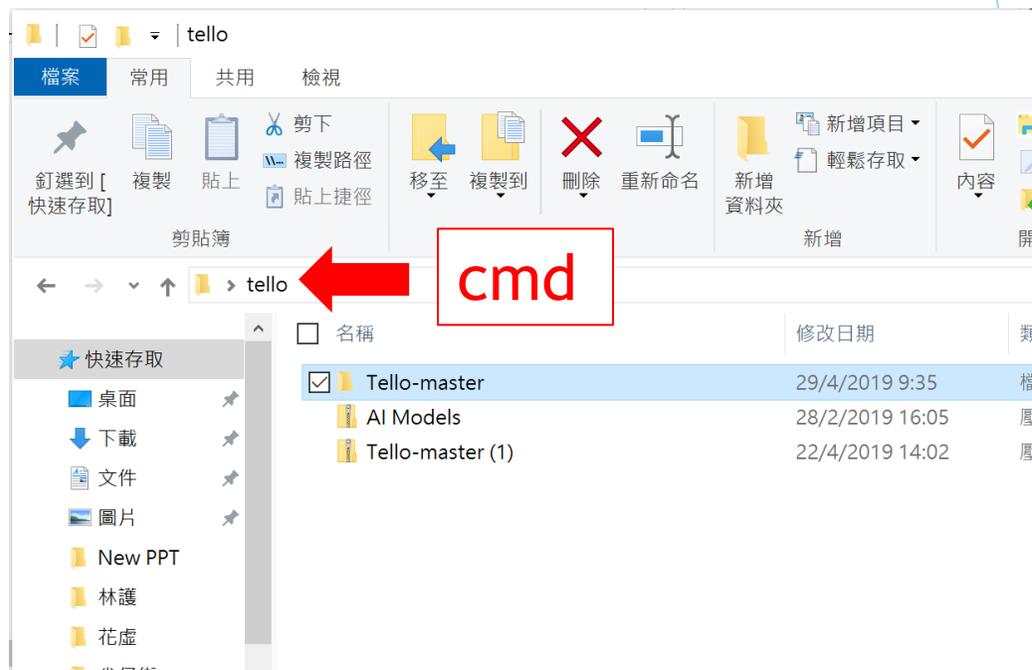


- ▶ Visual Studio Code
- ▶ EditPlus
- ▶ UltraEdit
- ▶ PSPad
- ▶ Notepad++

利用Python連接Tello Edu

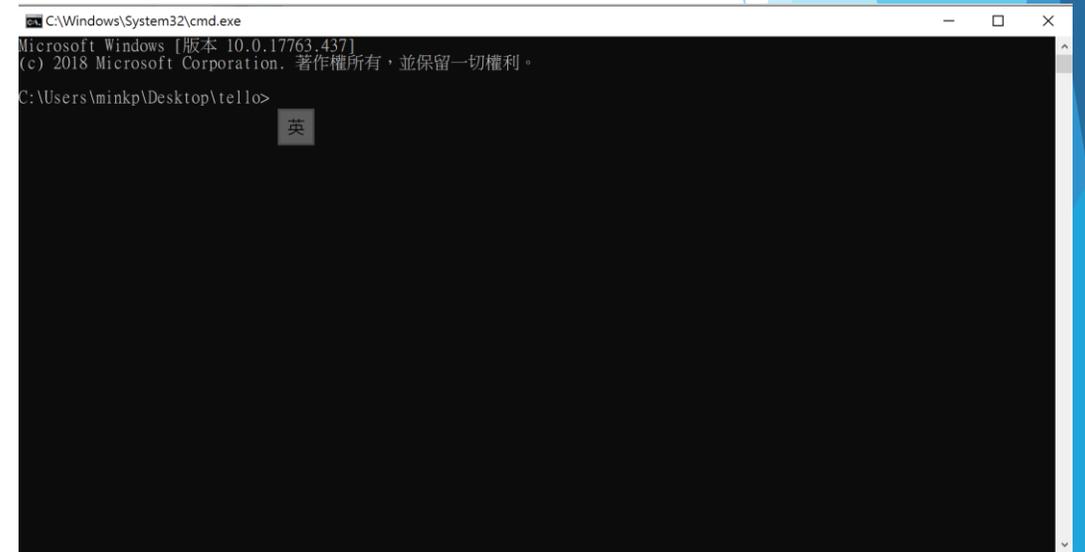
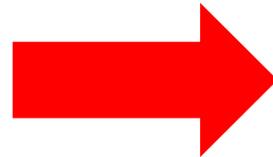
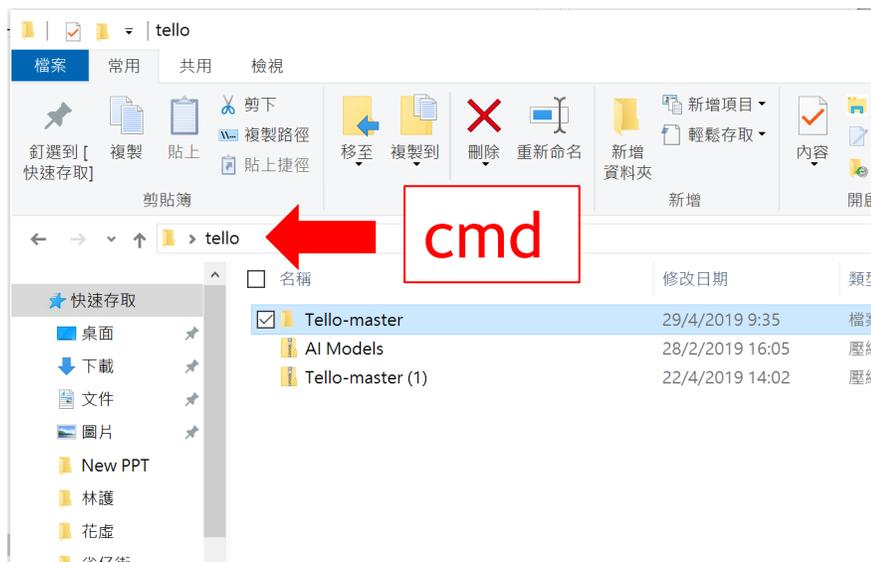


- ▶ 開啟Tello Edu 並連接電腦
- ▶ 開啟桌面上的Tello資料夾
- ▶ 輸入"cmd"



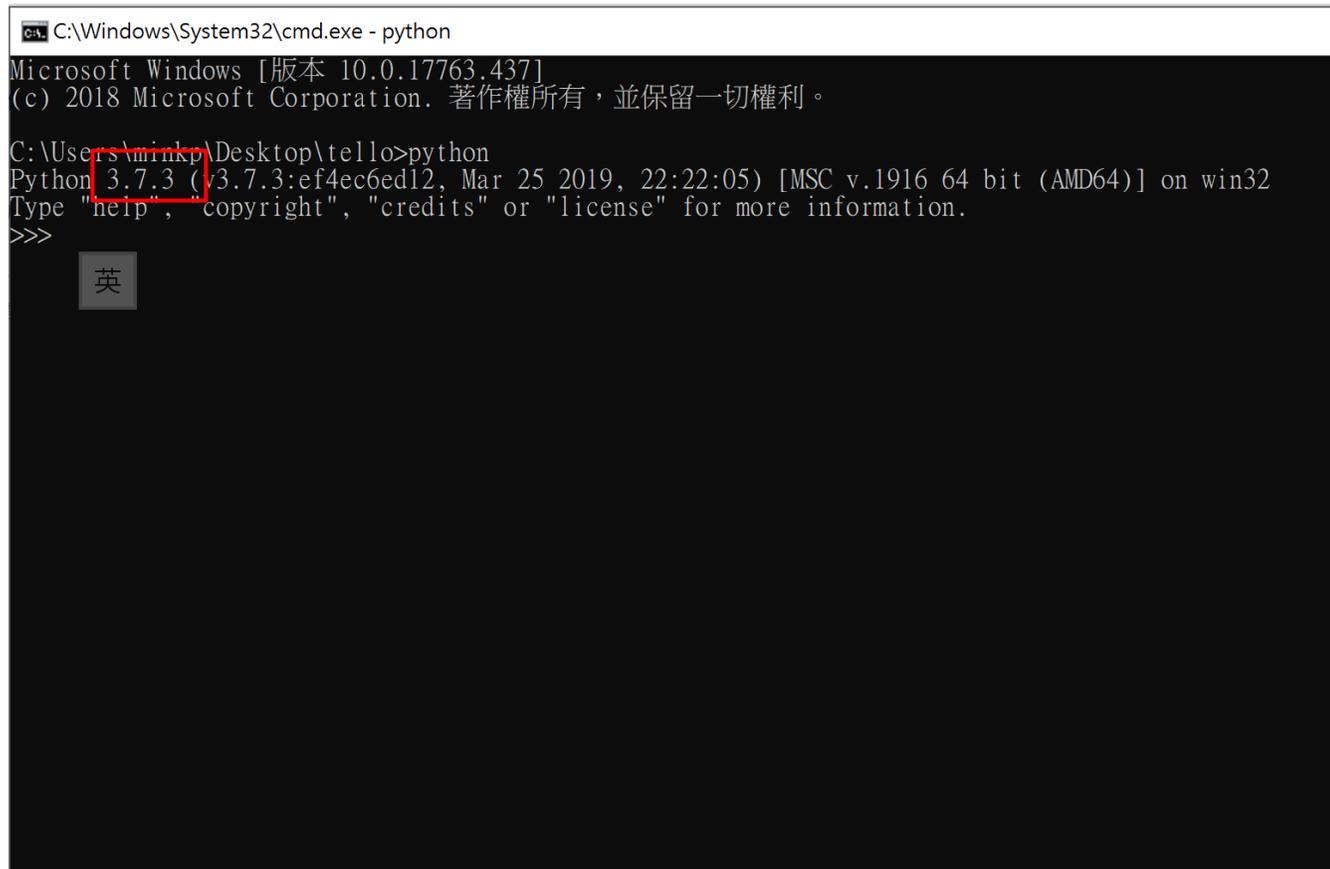
活動一：利用Python連接Tello Edu

- ▶ 開啟桌面上的Tello資料夾
- ▶ 輸入"cmd"



活動一：利用Python連接Tello Edu

- ▶ 輸入“python” enter - 可查閱Python版本



```
C:\Windows\System32\cmd.exe - python
Microsoft Windows [版本 10.0.17763.437]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\minkp\Desktop\tello>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

英

活動一：利用Python連接Tello Edu

- ▶ 利用Wifi 連接Tello Edu
- ▶ 在cmd狀態下輸入“python”及“空白鍵”，並把資料夾中的test1.py檔案拖進去。



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.437]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\minkp\Desktop\tello\新增資料夾\Tello-master>python C:\Users\minkp\Desktop\tello\新增資料夾\Tello-master\test1.py
```

- ▶ 按enter後，Tello Edu 便會開始運作

活動一：利用Python連接Tello Edu

- ▶ 利用Wifi 連接Tello Edu
- ▶ 在cmd狀態下輸入“python”及“空白鍵”，並把資料夾中的test1.py檔案拖進去。



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.437]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\minkp\Desktop\tello\新增資料夾\Tello-master>python C:\Users\minkp\Desktop\tello\新增資料夾\Tello-master\test1.py
```

- ▶ 按enter後，Tello Edu 便會開始運作

活動二：了解Python Code

- ▶ 利用 Visual Studio Code 打開 “test1.py”



```
File Edit Selection View Go Debug Terminal Help
Release Notes: 1.33.1 test1.py x
1 from tello import *
2
3 start()
4 takeoff()
5
6 result = get_tof()
7
8 print(result)
9
10 land() # Land the drone
11
```

開啟TOF高度感測器

讀取高度結果

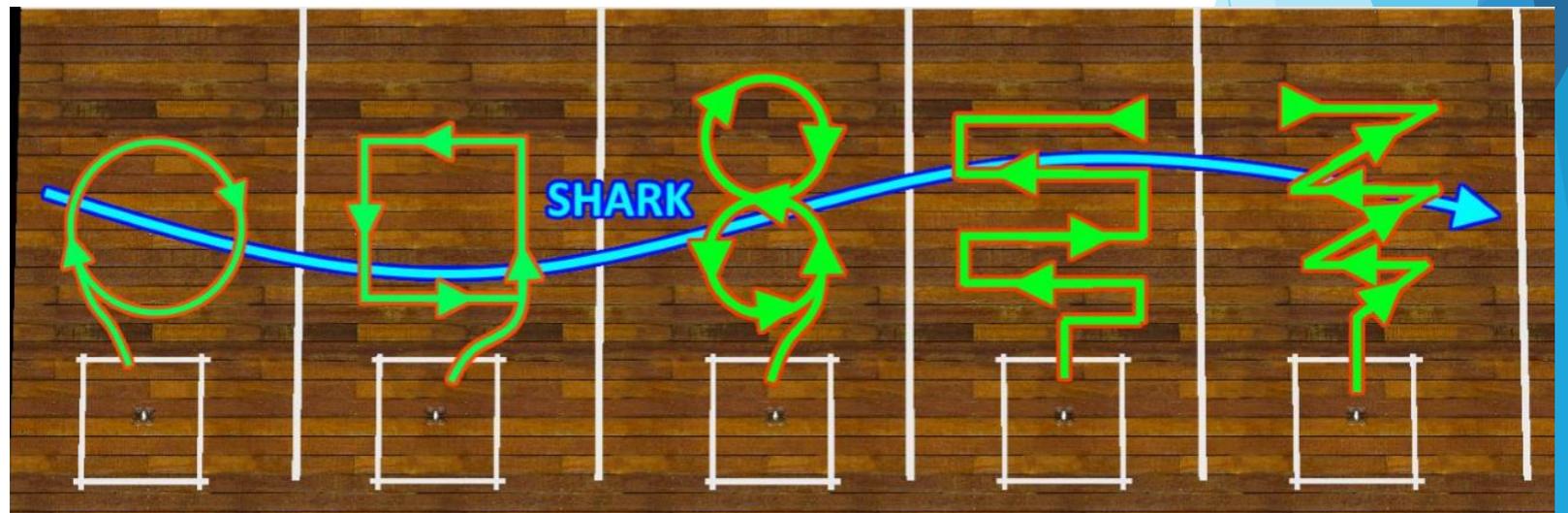
活動二：了解Python Code

- ▶ 重覆(repeat)：
 - ▶ 利用以下語法可重覆各步驟：

```
for i in range(5):
```

重覆指令

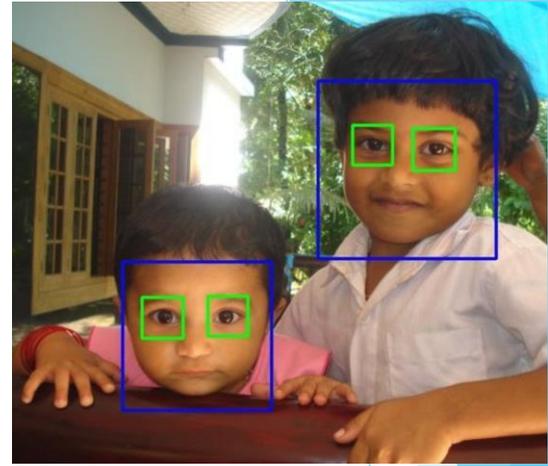
重覆次數



AI 人臉識別原理

人臉識別是由一系列的幾個相關問題組成的：

- 1、首先找到一張圖片中的所有人臉。
- 2、對於每一張臉來說，無論光線明暗或面朝別處，它依舊能夠識別出是同一個人的臉。
- 3、能夠在每一張臉上找出可用於他人區分的獨特之處，比如眼睛多大，臉有多長等等。
- 4、最後將這張臉的特點與已知所有人臉進行比較，以確定這個人是誰。

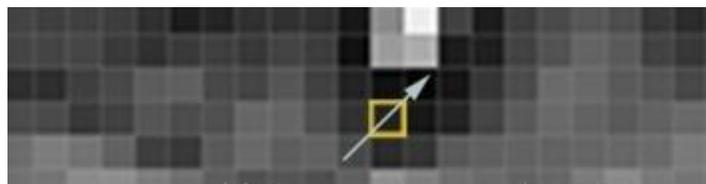
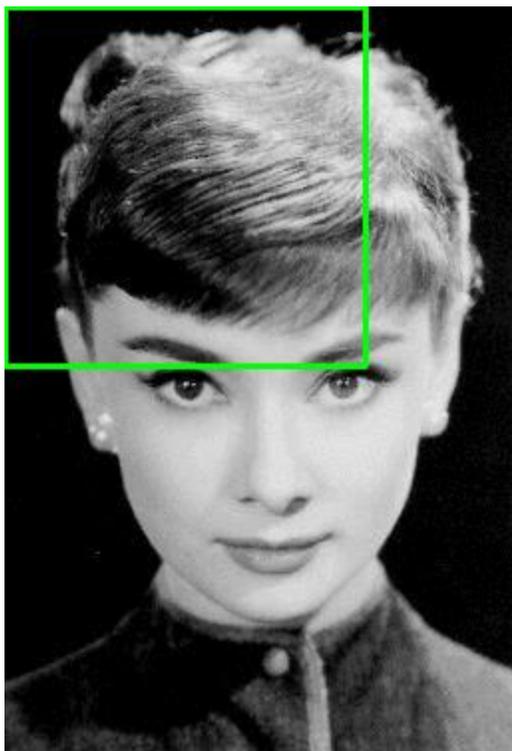


OpenCV

OpenCV (Open Source Computer Vision) 是一套強大且免費商業化開放式授權原始碼，同時也是一套跨平台支援各作業系統之電腦視覺資料庫。

OpenCV 專案最早由英特爾公司於 1999 年創立，主要是為了推廣機器視覺的研究，提供一套開源且最佳化的基礎，裡面包含許多影像處理、機械學習演算法及電腦視覺(Computer Vision)應用所需的函式庫(Library)。

OpenCV 原理



```
0.097496084868908 0.045223236083984 -0.1281466782093 0.032084941864014
0.12529824674129 0.060309179127216 0.17521631717682 0.020976085215807
0.030809439718723 -0.01981477253139 0.10801389068365 -0.00052163278451189
0.036050599068403 0.065554238855839 0.0731306001544 -0.1318951100111
-0.097486883401871 -0.0066401711665094 0.1226262897253 -0.0059557510539889
-0.14131525158882 0.036750309169292 0.036750309169292 0.043374512344599
-0.048540540039539 -0.1061901587992907 -0.15042643249035 -0.053343612700701
-0.12567175924778 -0.10568545013666 -0.12728653848171 -0.076289616525173
-0.061418771743774 -0.074287034571171 -0.065365232527256 0.12369467318058
0.046741496771574 0.0061761881224811 0.14746543765068 0.056418422609568
-0.12113650143147 -0.21055991947651 0.0041091227903962 0.089727647602558
0.061606746166945 0.11345765739679 0.021352224051952 -0.0085843298584223
0.061989940702915 0.19372203946114 -0.086726233363152 -0.022388197481632
0.10904195904732 0.084853030741215 0.09463594853878 0.020696049556136
-0.019414527341723 0.0064811296761036 0.21180312335491 -0.050584398210049
0.15245945751667 -0.16582328081131 -0.035577941685915 -0.072376452386379
-0.12216668576002 -0.007277755558491 -0.036901291459799 -0.034365277737379
0.083934605121613 -0.059730969369411 -0.070026844739914 -0.045013956725597
0.087945111095905 -0.11478432267904 -0.089621491730213 -0.013955107890069
-0.021407851949334 0.14841195940971 0.078333757817745 -0.17898085713387
-0.018298890441656 0.049525424838066 0.13227833807468 -0.072600327432156
-0.011014151386917 -0.051016297191381 -0.14132921397686 0.0050511928275228
0.0093679334968328 -0.062812767922878 -0.13407498598099 -0.014829395338893
0.058139257133007 0.0048638740554452 -0.039491076022387 -0.043765489012003
-0.024210374802351 -0.11443792283535 0.071997955441475 -0.012062266469002
-0.057223934680223 0.014683869667351 0.05228154733777 0.012774495407939
0.023535015061498 -0.081752359867096 -0.031709920614958 0.069833360612392
-0.0098039731383324 0.037022035568953 0.11009479314089 0.11638788878918
0.020220354199409 0.12788131833076 0.18632389605045 -0.015336792916059
0.0040337680839002 -0.094398014247417 -0.11768248677254 0.10281457751989
0.051597066223621 -0.10034311562777 -0.040977258235216 -0.082041338086128
```

識別結果

$$d(x, y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

活動三：視頻模式

- ▶ 利用以下語法可開啟視頻模式：

```
start_video() # Start the video stream
```

```
time.sleep(30) # Wait for 30 seconds
```

```
stop_video() # Stop the video stream
```

活動四：AI 人臉識別範例

- ▶ 利用Visual Studio Code開啟Eye_See_You

```
eyeCascade = cv2.CascadeClassifier('haarcascade_eye.xml')
```

使用Haar Cascades進行人臉檢測



使用Haar Cascades進行人臉檢測

Data Size: 2MB



Data Size: 1.3MB



使用Haar Cascades進行人臉檢測

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
eyes = eyeCascade.detectMultiScale(gray, scaleFactor = 1.1 ,minNeighbors = 5 ,minSize = ( 30 , 30 ),flags = cv2 . CASCADE_SCALE_IMAGE)
print ( 'eyes found: ' + str(len (eyes)))
```

活動四：AI 人臉識別範例

- ▶ 利用Visual Studio Code開啟Eye_See_You

```
print('eyes found!')
if len(eyes) >= 4:
    up(50)
    down(50)
right(50)
left(50)
time.sleep(1)
```

謝謝參與
無人機中學教師培訓

