

序

Python 語言是 Guido Van Rossum 開發的一種通用用途（General Purpose）的程式語言，這是擁有優雅語法和高可讀性程式碼的程式語言，可以讓我們開發 GUI 視窗程式、Web 應用程式、系統管理工作、財務分析、大數據資料分析和人工智慧等各種不同的應用程式。

人工智慧（Artificial Intelligence，AI）是讓機器變得更聰明的一種科技，可以讓機器具備和人類一樣的思考邏輯與行為模式。機器學習就是一種人工智慧，對於初學者來說，你不用自行訓練模型，一樣可以學習人工智慧程式設計，使用預訓練模型來建立各種人工智慧的相關應用。

本書是一本學習 Python 程式設計的入門教材，也是一本入門人工智慧程式設計的基礎教材，可以讓初學者輕鬆自行建立相關的人工智慧應用。在規劃上，本書可以作為大學、科技大學和技術學院 Python 程式設計，或人工智慧入門課程的教課書，適用 3 學分一個學期或 2 學分二個學期課程的上課教材。

在內容上本書不只完整說明人工智慧世代的你需要具備的 Python 程式設計能力，更詳細說明常見人工智慧應用的相關套件，可以讓你馬上靈活運用這些套件來建立你自己的人工智慧應用，包含：物體識別和 OCR 文字識別，人臉、多手勢追蹤、姿勢偵測（即時偵測出人臉、手勢和姿勢），與人臉識別（辨識出這是誰的臉），最後實際運用 Python 來建立剪刀、石頭、布等手勢操控 Windows 應用程式、AI 健身教練、車牌辨識和刷臉點名/報到等專案開發。

不只如此，為了方便初學者學習基礎結構化程式設計，本書使用大量圖例和流程圖來詳細說明程式設計的觀念和語法，在流程圖部分是使用 fChart 流程圖直譯器，此工具不只可以繪製流程圖，更可以使用動畫執行流程圖來驗證程式邏輯的正確性，讓讀者學習使用電腦的思考模式來撰寫 Python 程式碼，完整訓練和提昇你的邏輯思考、抽象推理與問題解決能力。

最後，實際使用 Teachable Machine 網頁工具說明訓練 AI 模型的步驟和過程，讓你一樣可以輕鬆建立自己專屬的深度學習模型與應用。

編著本書雖力求完美，但學識與經驗不足，謬誤難免，尚祈讀者不吝指正。

陳會安於台北 hueyan@ms2.hinet.net

2022.5.30

■ 範例檔說明

為了方便讀者學習Python人工智慧程式設計，筆者已經將本書的Python範例程式和相關檔案都收錄在書附範例檔，如下表所示：

檔案與資料夾	說明
ch01~ch10、ch12~ch16和cvzone資料夾	本書各章 Python範例程式、測試圖檔、視訊檔和預訓練模型等相關檔案
本書各章 pip安裝的套件清單.txt	本書各章 pip安裝套件的 Python版本和命令列指令

在fChart流程圖教學工具的官方網站，可以下載配合本書使用的WinPython客製化Python套件（請在上方選【Python套件】標籤頁，可以看到本書書名和列出Python套件的下載超連結，請任選一個下載），如下所示：



- <https://fchart.github.io/>

因為Anaconda整合散發套件和Python套件的改版十分頻繁，為了方便讀者練習和學校上課教學所需（避免版本不相容問題），本書提供整合fChart的客製化WinPython套件的可攜式Python開發環境，只需下載執行和解壓縮後，就可以建立執行本書Python程式和Thonny整合開發環境。

在客製化WinPython套件已經安裝好Thonny和IDLE和執行本書Python程式所需的套件，為了方便啟動相關工具，更提供工作列的「fChart主選單」可以快速啟動相關工具。

■ 版權聲明

本書範例檔案提供的共享軟體或公共軟體，其著作權皆屬原開發廠商或著作人，請於安裝後詳細閱讀各工具的授權和使用說明。在本書範例檔內含的軟體和媒體檔都為隨書贈送，僅提供本書讀者練習之用，與各軟體和媒體檔的著作權和其它利益無涉，如果使用過程中因軟體所造成的任何損失，與本書作者和出版商無關。

目錄

01/ Python語言與運算思維基礎

- 1-1 程式與程式邏輯 1-2
- 1-2 認識Python、運算思維和Thonny 1-5
- 1-3 下載與安裝Thonny 1-8
- 1-4 使用Thonny建立第一個Python程式 1-10
- 1-5 Thonny基本使用與程式除錯 1-14

02/ 寫出和認識Python程式

- 2-1 開發Python程式的基本步驟 2-2
- 2-2 編輯現存的Python程式 2-3
- 2-3 建立第二個Python程式的加法運算 2-7
- 2-4 看看Python程式的內容 2-11
- 2-5 Python文字值 2-12
- 2-6 Python寫作風格 2-17

03/ 變數、運算式與運算子

- 3-1 程式語言的變數 3-2
- 3-2 在程式使用變數 3-5
- 3-3 變數的資料型態和型態轉換函數 3-9
- 3-4 讓使用者輸入變數值 3-11
- 3-5 認識運算式和運算子 3-12
- 3-6 在程式使用運算子 3-17

04 條件判斷

4-1	你的程式可以走不同的路	4-2
4-2	關係運算子與條件運算式	4-4
4-3	if單選條件敘述	4-7
4-4	if/else二選一條件敘述	4-12
4-5	if/elif/else多選一條件敘述	4-16
4-6	在條件敘述使用邏輯運算子	4-19

05 重複執行程式碼

5-1	認識迴圈敘述	5-2
5-2	for計數迴圈	5-3
5-3	while條件迴圈	5-10
5-4	改變迴圈的執行流程	5-14
5-5	巢狀迴圈與無窮迴圈	5-18
5-6	在迴圈中使用條件敘述	5-22

06 函數

6-1	認識函數	6-2
6-2	使用者自訂函數	6-3
6-3	函數的參數	6-7
6-4	函數的回傳值	6-14
6-5	函數的實際應用	6-18
6-6	變數範圍和內建函數	6-21

07 字串與容器型態

7-1	字串型態	7-2
7-2	串列型態	7-6
7-3	元組型態	7-13
7-4	字典型態	7-16
7-5	字串與容器型態的運算子	7-21



認識 Python、運算思維和 Thonny

我們學習程式設計的目的是訓練你的運算思維，在本書是使用 Thonny 整合開發環境來學習 Python 人工智慧程式設計。

1-2-1 談談運算思維與演算法

如同建設公司興建大樓有建築師繪製的藍圖，廚師烹調有食譜，設計師進行服裝設計有設計圖，程式設計也一樣有藍圖，那就是演算法。運算思維最重要的部分就是演算法。

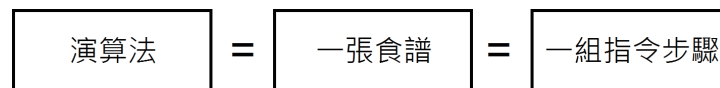
📍 運算思維

對於身處資訊世代的我們來說，運算思維 (Computational Thinking) 被認為這一世代必備的核心技能，不論你是否是資訊相關科系的學生或從事此行業，運算思維都可以讓你以更實務的思維來看這個世界。基本上，運算思維可以分成五大領域，如下所示：

- ▷ 抽象化 (Abstraction)：思考不同層次的問題解決步驟。
- ▷ 演算法 (Algorithms)：將解決問題的工作思考成一序列可行且有限的步驟。
- ▷ 分割問題 (Decomposition)：了解在處理大型問題時，我們需要將大型問題分割成小問題的集合，然後個個擊破來一一解決。
- ▷ 樣式識別 (Pattern Recognition)：察覺新問題是否和之前已解決問題之間擁有關係，可以讓我們直接使用已知或現成的解決方法來解決問題。
- ▷ 歸納 (Generalization)：了解已解決的問題可能是用來解決其他或更大範圍問題的關鍵。

📍 演算法

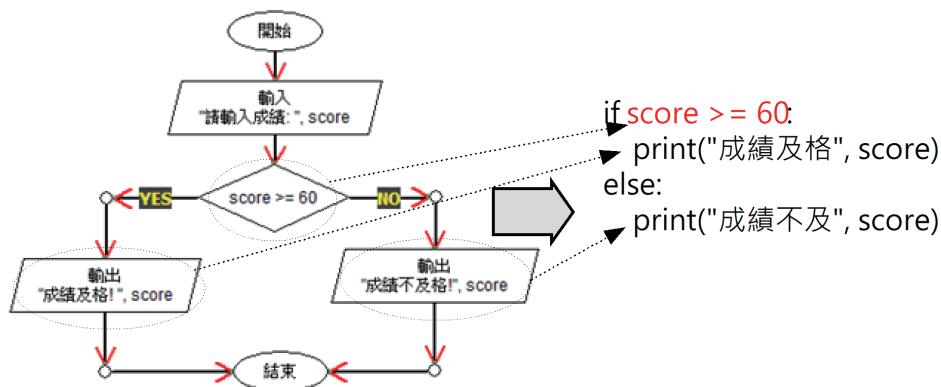
演算法 (Algorithms) 簡單的說就是一張食譜 (Recipe)，提供一組一步接著一步 (Step-by-step) 的詳細過程，包含動作和順序，可以將食材烹調成美味的食物，例如：在第 1-1-1 節說明的蛋糕製作，製作蛋糕的食譜就是一個演算法，如下圖所示：



電腦科學的演算法是用來描述解決問題的過程，也就是完成一個任務所需的具體步驟和方法，這個步驟是有限的；可行的，而且沒有模稜兩可的情況。

📍 使用流程圖描述演算法

演算法可以使用文字描述或圖形化方式來描述，圖形化方式就是流程圖 (Flow Chart)，流程圖是使用標準圖示符號來描述執行過程，以各種不同形狀的圖示表示不同的操作，箭頭線標示流程執行的方向，當畫出流程圖的執行過程後，就可以轉換撰寫成特定語言的程式碼，例如：Python 語言，如下圖所示：

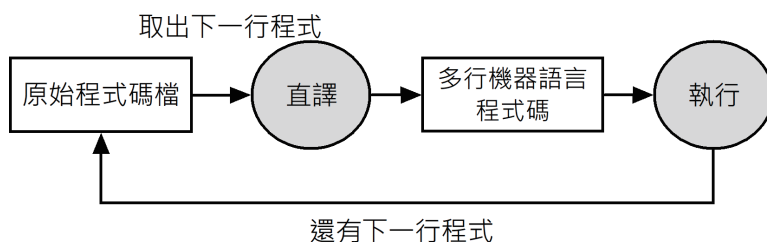


1-2-2 認識 Python 語言

Python 語言是 Guido Van Rossum 開發的一種通用用途 (General Purpose) 的程式語言，這是擁有優雅語法和高可讀性程式碼的程式語言，可以讓我們開發 GUI 視窗程式、Web 應用程式、系統管理工作、財務分析、大數據資料分析和人工智慧等各種不同的應用程式。

Python 語言兩個版本：Python 2 和 Python 3，在本書說明的是 Python 3 語言，其特點如下所示：

- ▷ Python 是一種直譯語言 (Interpreted Language)：Python 程式是使用直譯器 (Interpreters) 來執行，直譯器並不會輸出可執行檔案，而是一個指令一個動作，一行一行原始程式碼轉換成機器語言後，馬上執行程式碼，如下圖所示：



- ▷ Python 是動態型態 (Dynamically Typed) 語言：Python 變數並不需要預先宣告資料型態，Python 直譯器會依據變數值來自動判斷資料型態。當 Python 程式碼將變數 **a** 指定成整數 1，變數的資料型態是整數；變數 **b** 指定成字串，資料型態就是字串，如下所示：

```
a = 1
b = "Hello World!"
```

- ▷ Python 是強型態 (Strongly Typed) 語言：Python 並不會自動轉換變數的資料型態，當 Python 程式碼是字串加上整數，因為 Python 不會自動型態轉換，我們需要自行使用 **str()** 函數轉換成同一型態的字串，否則就會產生錯誤，如下所示：

```
"計算結果 = " + 100      # 錯誤寫法
"計算結果 = " + str(100)  # 正確寫法
```

1-2-3 Thonny 整合開發環境

雖然使用純文字編輯器，例如：記事本，就可以輸入 Python 程式碼，但是對於初學者來說，建議使用「IDE」(Integrated Development Environment) 整合開發環境來學習 Python 程式設計，「開發環境」(Development Environment) 是一種工具程式，可以用來建立、編譯 / 直譯和除錯指定程式語言所建立的程式碼。

目前高階程式語言大都有提供整合開發環境，可以在同一工具來編輯、編譯 / 直譯和執行特定語言的程式。Thonny 是愛沙尼亞 Tartu 大學開發，一套完全針對「初學者」開發的免費 Python 整合開發環境，其主要特點如下所示：

- ▷ Thonny 支援 Python 和 MicroPython 語言。
- ▷ Thonny 支援自動程式碼完成和括號提示，可以幫助初學者輸入正確的 Python 程式碼。
- ▷ Thonny 使用即時高亮度提示程式碼錯誤，並且提供協助說明和程式碼除錯，可以讓我們一步一步執程式碼來進程式碼除錯。



下載與安裝 Thonny

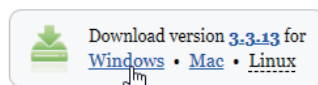
Thonny 跨平台支援 Windows、MacOS 和 Linux 作業系統，可以在 Thonny 官方網站免費下載最新版本（Thonny 本身就是使用 Python 開發）。

📍 方法一：在官網自行下載和安裝 Thonny

Thonny 可以在官方網站免費下載，其 URL 網址如下所示：

▷ <https://thonny.org/>

Thonny
Python IDE for beginners



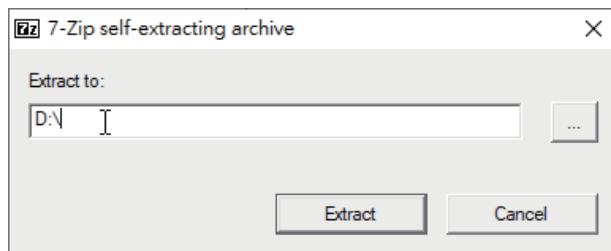
請點選【Windows】超連結下載最新版 Thonny 安裝程式，就可以在 Windows 電腦執行下載的安裝程式來安裝 Thonny。請注意！讀者需參閱第 9 章的說明自行安裝本書各章節所需的 Python 套件。

📍 方法二：下載安裝本書客製化 WinPython 可攜式套件

為了方便老師教學和讀者自學 Python 人工智慧程式設計，本書提供一套客製化 WinPython 套件的 Python 開發環境，已經安裝好 Thonny 和本書各章節使用的套件，只需解壓縮，就可以馬上建立可執行本書 Python 範例程式的開發環境。

請參閱書附範例檔的說明來下載客製化 WinPython 套件的 Python 開發環境，此套件是一個 7-Zip 格式的自解壓縮檔，下載檔名是：fChartThonny6_3.9AI.exe。

當成功下載套件後，請執行 7-Zip 自解壓縮檔，在【Extract to:】欄位輸入解壓縮的硬碟，例如：「C:\」或「D:\」等，按【Extract】鈕，就可以解壓縮安裝 WinPython 套件的 Python 開發環境，如下圖所示：





CHAPTER

12

人工智慧應用（一）： 人臉偵測與人臉識別

📌 本章內容

- 12-1 MediaPipe 框架與 CVZone 電腦視覺套件
- 12-2 使用 CVZone 人臉偵測
- 12-3 使用 CVZone 臉部網格
- 12-4 CVZone 辨識臉部表情：張嘴 / 閉嘴與開眼 / 閉眼
- 12-5 face-recognition 人臉識別



MediaPipe 框架與 CVZone 電腦視覺套件

MediaPipe 是 Google 公司開發的機器學習框架，這是跨平台的機器學習解決方案，可以讓 AI 研究者和開發者建立世界等級，針對手機、PC、雲端、Web 和 IoT 裝置的機器學習應用程式和解決方案。

MediaPipe 跨平台支援 Android、iOS、Web 和邊緣運算裝置，C++、JavaScript 和 Python 程式語言，隨著平台釋出的相關應用範例，可以讓我們馬上執行人工智慧應用，包含：人臉偵測 (Face Detection)、多手勢追蹤 (Multi-hand Tracking) 和人體姿態評估 (Human Pose Estimation) 等。

在 Python 開發環境安裝 MediaPipe 套件的命令列指令，如下所示：

```
pip install mediapipe==0.8.9.1 
```

● 說明 ●

請注意！當執行 MediaPipe 的 Python 程式時，如果出現下列錯誤訊息，如下所示：

```
from mediapipe.python._framework_bindings import resource_util
ImportError: DLL load failed: The specified module could not be found.
```

第一種解決方式是安裝 `msvc-runtime`，其命令列指令如下所示：

```
pip install msvc-runtime 
```

如果仍然產生錯誤訊息，請在 Windows 作業系統安裝 Microsoft Visual C++ 可轉散發套件，其下載網址：<https://docs.microsoft.com/zh-tw/cpp/windows/latest-supported-vc-redist?view=msvc-170>，下載檔名是 `vc_redist.x64.exe`。



CVZone 是基於 OpenCV 和 MediaPipe 的 Python 電腦視學套件，可以讓我們使用更少的程式碼，和以更容易的方式來輕鬆進行人臉偵測、3D 臉部網格、多手勢追蹤和人體姿態評估等電腦視學應用。

在 Python 開發環境安裝 CVZone 套件的命令列指令，如下所示：

```
pip install cvzone==1.5.3 
```

● 說明 ●

因為 MediaPipe 改版比 CVZone 快，CVZone 套件的部分模組並不支援新版 MediaPipe，所以筆者已經更新模組，同時新增一些取出內部資料的功能，在本書提供的 WinPython 套件已經更新這些模組。

如果讀者自行在 Python 開發環境安裝 CVZone 套件，可以有兩種方式來更新這些模組，如下所示：

- ▷ 方法一：複製書附範例檔「cvzone」目錄下的 4 個 Python 檔案(請注意！不包含 __init__.py)，即 FaceDetectionModule.py、FaceMeshModule.py、HandTrackingModule.py 和 PoseModule.py，然後取代 Python 開發環境的 CVZone 安裝目錄「python-3.9.8.amd64\Lib\site-packages\cvzone」下的同名 4 個檔案(以 3.9.8 版為例)。
- ▷ 方法二：將書附範例檔「cvzone」目錄整個複製到與 Python 程式檔案位在相同目錄，因為在「cvzone」子目錄有 __init__.py 檔(此目錄是套件)，Python 程式可以使用此目錄的套件取代 CVZone 套件，不過！此方法將無法使用 CVZone 套件的其他模組。



使用 CVZone 人臉偵測

CVZone 人臉偵測是基於 MediaPipe 人臉偵測 (Face Detection)，這是使用 BlazeFace 模型的一種超快速人臉偵測，可以在影像中偵測出多張人臉，和標示臉部的 6 個關鍵點 (Key Points)：左眼、右眼、鼻尖、嘴巴、左耳和右耳。

12-2-1 CVZone 人臉偵測的基本使用

在 Python 程式建立 CVZone 的 FaceDetector 物件後，就可以呼叫 `findFaces()` 方法來偵測人臉。

📍 影像的人臉偵測：ch12-2-1.py

在 Python 程式可以偵測 faces.jpg 影像的所有人臉，首先從 CVZone 的 FaceDetectionModule 模組匯入 FaceDetector 類別，和 OpenCV，如下所示：

```
from cvzone.FaceDetectionModule import FaceDetector
import cv2

img = cv2.imread("images/faces.jpg")
detector = FaceDetector(minDetectionCon=0.5)
```

上述程式碼讀取圖檔後，建立 `FaceDetector` 物件 `detector`，參數 `minDetectionCon` 是最低信心指數 (0~1)，預設值 0.5 是信心指數 (即可能性) 需超過 50% 可能性才視為是人臉，然後呼叫 `findFaces()` 方法偵測人臉，如下所示：

```
img, faces = detector.findFaces(img)
if faces:
    print("偵測到人臉數:", len(faces))
```

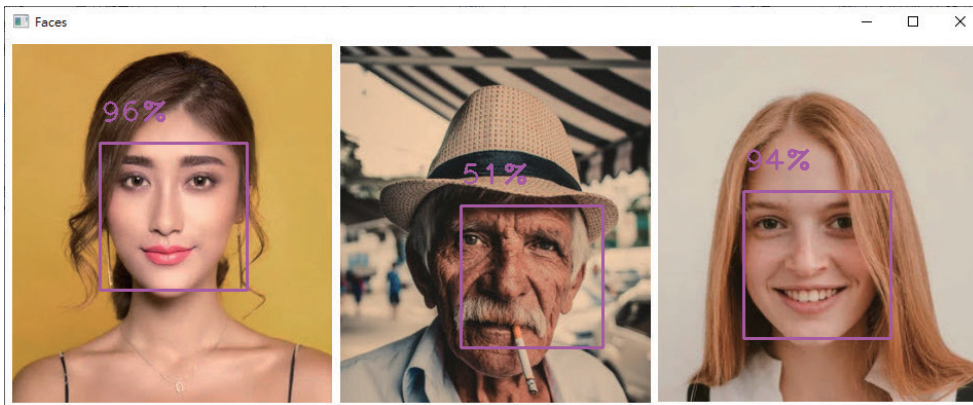
上述 `findFaces()` 方法的參數是影像內容，回傳值有 2 個，第 1 個是已經標示人臉方框和可能性的影像，第 2 個是偵測到的人臉資訊串列。然後使用 `if` 條件判斷是否偵測到人臉，有，就呼叫 `len()` 函數顯示偵測到的人臉數。在下方顯示已標示人臉方框和可能性的影像，如下所示：

```
cv2.imshow("Faces", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Python 程式的執行結果可以看到共偵測到 3 張臉，如下所示：

```
>>> %Run ch12-2-1.py
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
偵測到人臉數： 3
```

然後，可以看到紅色方框標示的人臉和上方可能性的百分比，如下圖所示：



如果不需要在影像上標示人臉方框和可能性百分比，請指定 `draw` 參數值 `False` (Python 程式：ch12-2-1a.py)，如下所示：

```
img, faces = detector.findFaces(img, draw=False)
```

📍 取得人臉的相關資訊：ch12-2-1b.py

在 `findFaces()` 方法的回傳值有偵測到人臉的相關資訊，因為偵測出的是多張臉，所以這是一個串列，第 1 張人臉是 `faces[0]`；第 2 張是 `faces[1]`，以此類推，如下所示：

```
img = cv2.imread("images/faces.jpg")
detector = FaceDetector()
img, faces = detector.findFaces(img)
if faces:
    print("偵測到人臉數:", len(faces))
    face = faces[0]
    print("id:", face["id"])
    print("bbox:", face["bbox"])
    print("score:", face["score"])
    print("center:", face["center"])
```

上述 `findFaces()` 方法偵測人臉和顯示人臉數後，使用 `faces[0]` 取得第 1 張臉，這是一個字典，"id" 是編號（從 0 開始），"bbox" 是方框座標的元組 (x, y, w, h)，分別是左上角座標 (x, y)、寬度和高度，"score" 是信心指數的可能性，最後的 "center" 是方框的中心點座標，其執行結果如下所示：

```
>>> %Run ch12-2-1b.py
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
偵測到人臉數: 3
id: 0
bbox: (92, 102, 144, 144)
score: [0.9604063630104065]
center: (164, 174)
```

上述可能性值是 0~1 之間的串列（只有 1 個元素），我們需要使用 `face["score"]` [0] 取出值後，轉換成整數的百分比（Python 程式：ch12-2-1c.py），如下所示：

```
int(face["score"][0] * 100)
```

12-2-2 顯示臉部關鍵點和剪裁出人臉

在 CVZone 原始版本並無法取得臉部 6 個關鍵點的座標，即左眼、右眼、鼻尖、嘴巴、左耳和右耳，筆者已經改寫模組新增此功能。在影像中如果需要剪裁出臉部影像，就是使用 NumPy 切割運算子。

學習評量

1. 請問什麼是 MediaPipe ? CVZone 套件的用途 ?
2. 請簡單說明人臉偵測和人臉識別的差異 ?
3. 請一一列出 face-recognition 套件的功能 ? 如何在 Windows 電腦的 Python 開發環境安裝此套件 ?
4. 請建立 Python 程式開啟 Webcam , 可以計算目前影格中有多少人 , 並且顯示在影格左上角 (提示 : 即偵測出的人臉數) 。
5. 請建立 Python 程式判斷臉部表情的右眼是張開或閉著 。
6. 請找出你自己的 3 張大頭貼照片 , 然後建立 Python 程式產生這 3 張照片的臉部資料編碼 , 然後計算出哪兩張照片的差異最小 。