期末專題報告

課程名稱: 人工智慧

照片分類模型

**開發語言、工具：Android studio Kotlin 、深度學習、**

**影像處理、資料庫設計**

S0854045 資工四 曾筠惠

民國110年6月19日

內容

[需求分析 3](#_Toc138613636)

[設計說明 4](#_Toc138613637)

[系統架構 7](#_Toc138613638)

[開發流程 7](#_Toc138613639)

[程式碼 12](#_Toc138613644)

[成果 16](#_Toc138613645)

[總結心得 19](#_Toc138613647)

# 需求分析

**說明:**

IG上看到很多網美的版面都是同一個色調，例如: **冷色調、暖色調等等**，版面乾淨、整齊可以上傳照片，而這些照片是怎麼從手機相簿幾千張照片找出來的呢，要如何從幾千張照片找到那一張接近版面色調的照片呢?

|  |  |  |
| --- | --- | --- |
| **常見IG色調** | | |
|  |  |  |
| **冷色調** | **暖色調** | **白色調** |

此外我發現版面看起來會如此整齊原因有三點 分別是1.物件 2.背景 3.調色

可以利用同一色系物件(藍色包包)，背景(藍色大海)去達到統一色調的效果，再不行就是調色，將白色的雪調成偏藍色。

|  |  |  |
| --- | --- | --- |
| **有哪些特徵** | | |
|  |  |  |
| 物件 | 背景 | 調色 |

根據上述觀察，我希望可以利用人工智慧，做出可以分類色彩的模型，並結合APP實作介面。

# 設計說明

照片分類模型

**目標:** 1.可以分類照片色彩的模型

**用途:** 1.方便使用者整理照片2.社群網站版面/發文

主要分成兩方向探討:

1. **顏色偵測**

**功能:** 擷取圖片實際顏色

**作法:** 使用K-means 分群方法

1. **色系分類**

**功能:** 將圖片實際顏色歸類到我想分類的色系

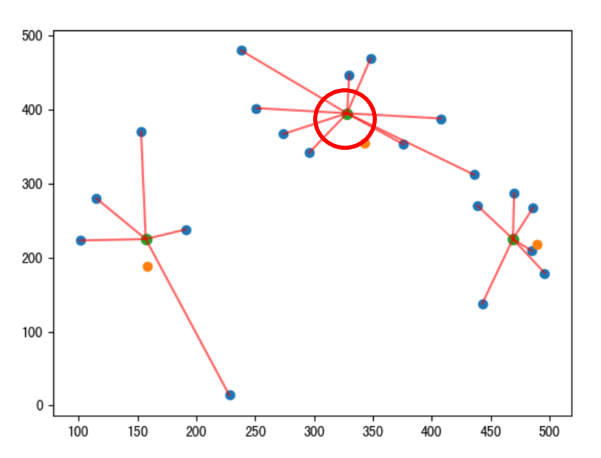
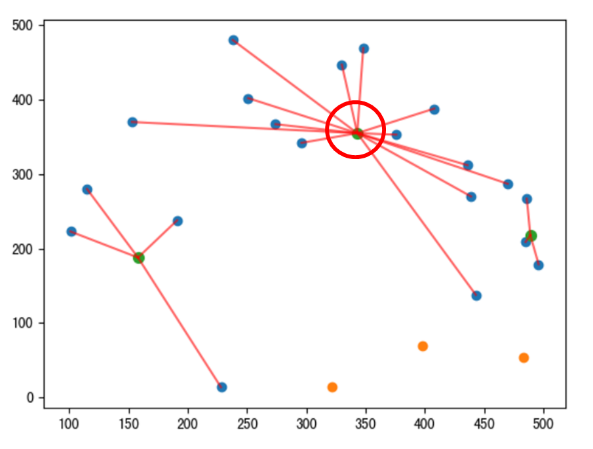
**作法:** 使用RGB threshold 將RGB色彩分類成gray、warm 、cool三類

1. **顏色偵測**

* **K-means 分群步驟:**

1. 先決定要分k組，並隨機選k個點做群集中心。
2. 將每一個點分類到離自己最近的群集中心(直線距離)。
3. 重新計算各組的群集中心(常用平均值)。

反覆 2、3 動作，直到群集不變，群集中心不動為止。



黃色是上一步群集中心的初始點，綠色為新的群集中心。

* **為什麼使用深度學習訓練?**

1. **可處理高維度的數據**

顏色由多個特徵或屬性組成，如RGB（紅綠藍）值或HSV（色相、飽和度、亮度）值。

1. **自動學習特徵**

傳統的機器學習方法需要手動提取特徵，可能無法完全捕捉顏色的差異。深度學習模型可以學習到較佳的特徵表示，更好區分不同的顏色。

1. **處理非線性關係**

顏色之間的關係是非線性的，傳統的機器學習方法難以捕捉到這種複雜的關係

* **為什麼選擇 K-means 模型?**

1. **簡單而直觀**

K-means是一種簡單且易於理解的聚類算法，可以將每個像素視為一個數據點，並使用K-means算法將像素分為K個不同的顏色集群。

1. **計算效率高**

相對於某些複雜的深度學習算法，K-means算法的計算效率相對較高，對大型圖像數據集處理來說很重要。

1. **可解釋性強(彈性高)**

K-means生成的結果易於解釋和理解，且可以調整 k值，根據自己的需求對結果進行後續處理。

因為我的設計是一次只需分類一張照片，且延遲不能太久，所以選擇高效率且簡單的K-means模型。

**2.色系分類**

我將色系分成三類 **1.灰階 2.冷色調 3.暖色調** ，並嘗試將顏色RGB值歸類到此三類。

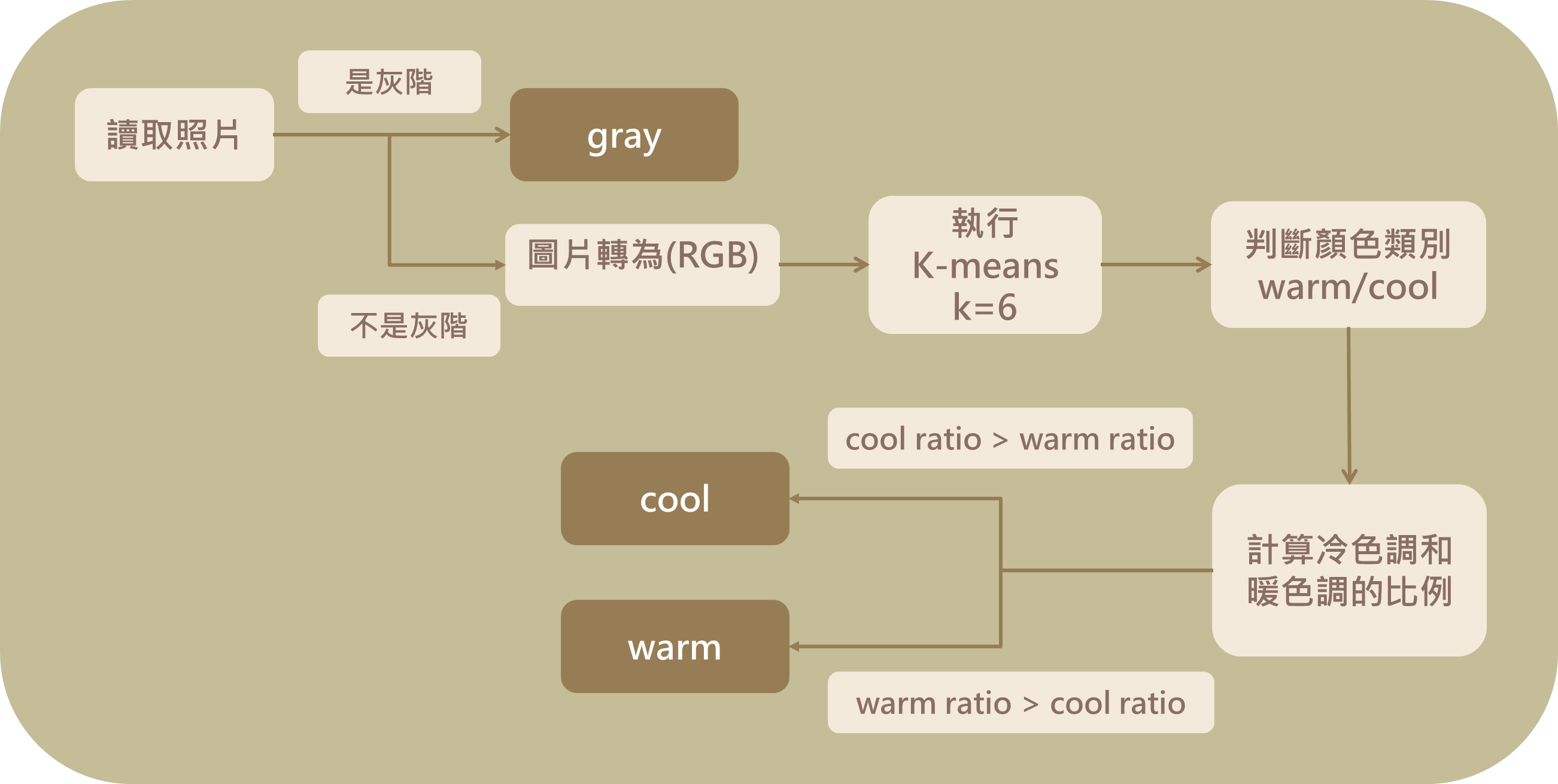
透過下表RGB 觀察

|  |  |  |
| --- | --- | --- |
| **灰階** | 冷色調 | 暖色調 |
|  |  |  |
| **R=G=B** | **(55, 64, 213)**  **(64, 213, 104)** | **(213, 64, 64)**  **(213, 153, 64)** |

觀察後發現

冷色R小 暖色R大，我根據這些特徵寫出判斷式

# 系統架構

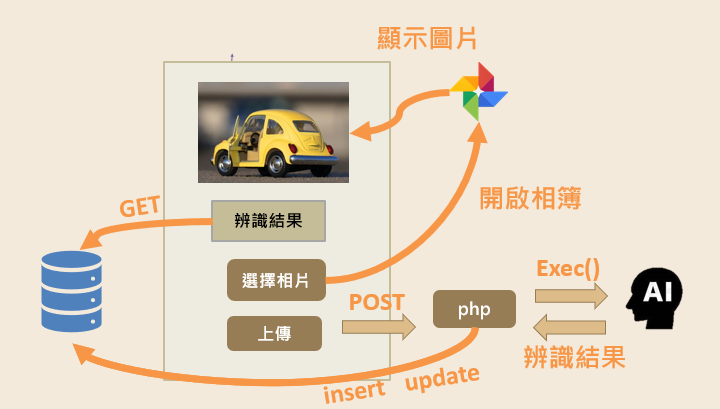


# 開發流程

* **與APP結合**

APP在接收到圖片後會呼叫python執行K-means分類

操作流程圖:

****

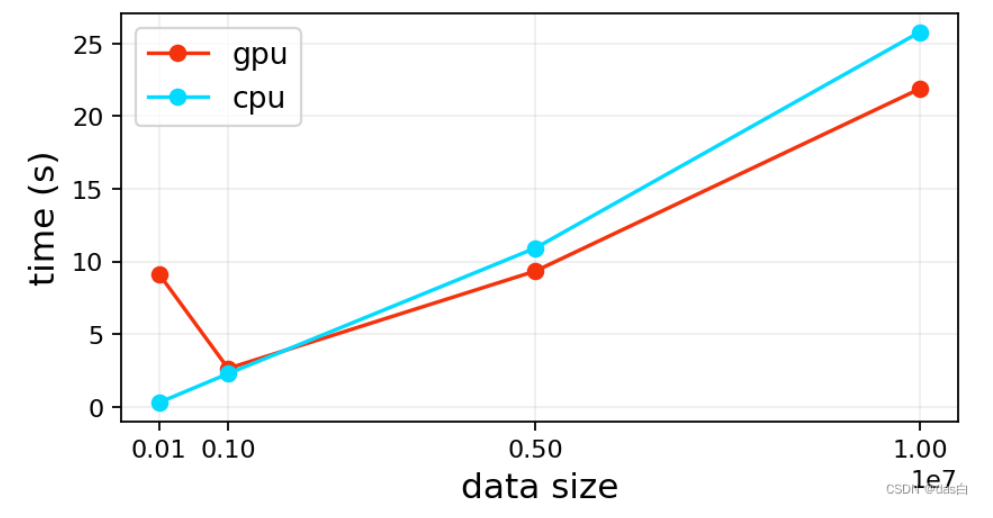
* K-means分群

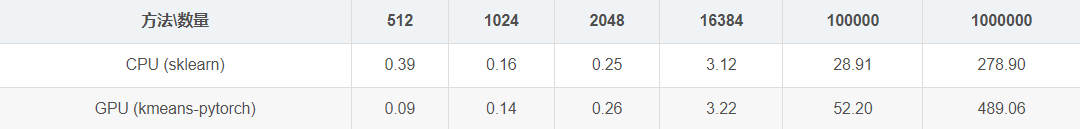
**說明:**

* 將圖片的像素數據（一維數組RGB值）作為輸入，為向量空間中的一個點。
* 將向量空間中的每個點分群，由使用者指定分成k群，便會有k個質心。(k=6)
* 顏色相近的點，在空間中的距離接近，會被分到同一質心，形成同一群。
* 最後重新計算6個群集，每一群集的質心位置（質心位置就是一組RGB位置），得到代表色與資料點數量，用圓餅圖呈現。
* **K-means module 選擇**

|  |  |  |
| --- | --- | --- |
| 方法 | 硬體 | 比較 |
| **Scikit-learn** | cpu | 優點：  適合進行簡單分類任務  劣勢：  當數據量較大時，迭代速度相對較慢 |
| 網路上人家寫的  kmeans function | gpu | 精度较差 |
| **kmeans-pytorch** | gpu | 能解决迭代速度問題  與sklearn相同的精度结果 |

CPU 和GPU 執行 k-means module 比較表:





**說明:**

可以看到GPU在數量多時 因為要CPU GPU來回轉換所以顯得比較慢。

我在測試的時候也是因為GPU要切換所以很慢，且因為我需求只需要讀一張照片CPU還可以負荷，所以選擇CPU。

* **灰階判斷**

**說明:**

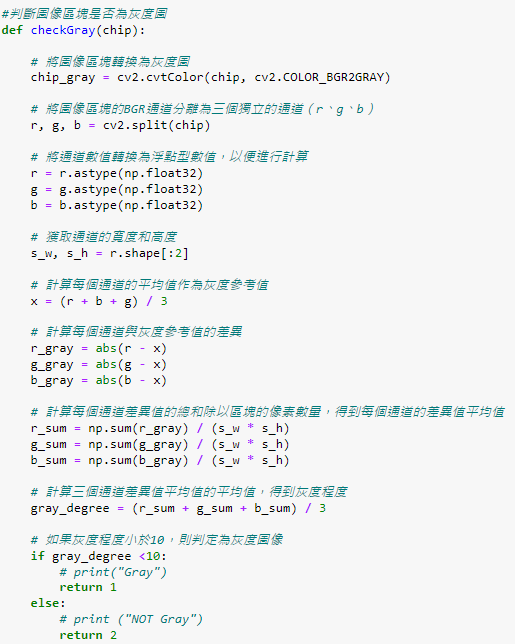
剛剛圖片有提到灰度圖中，RGB三個通道的值是相等的，即R=G=B，並且這些值的大小反映了灰度的深淺，從白色到黑色。

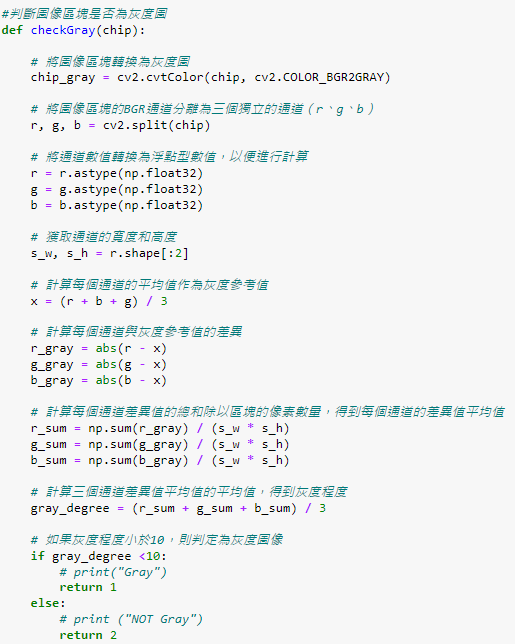
然而，僅僅通過檢查RGB圖像中的三個通道值是否相等是無法判斷區塊是否為灰度圖的。因為即使三個通道值不完全相等，但差異不大的情況下，對於人眼來說也可以視為灰度圖。

**步驟:**

* 獲取圖像區塊的像素值。
* 計算每個像素點的RGB通道平均值，即計算R、G、B的平均值。
* 將每個R、G、B的值與平均值進行比較，計算它們的差異。
* 判斷差異值是否小於某個閾值。如果差異值小於閾值(10)，則認為該區塊是灰度圖；否則，認為該區塊不是灰度圖。

**程式碼:**





* 暖/冷色系

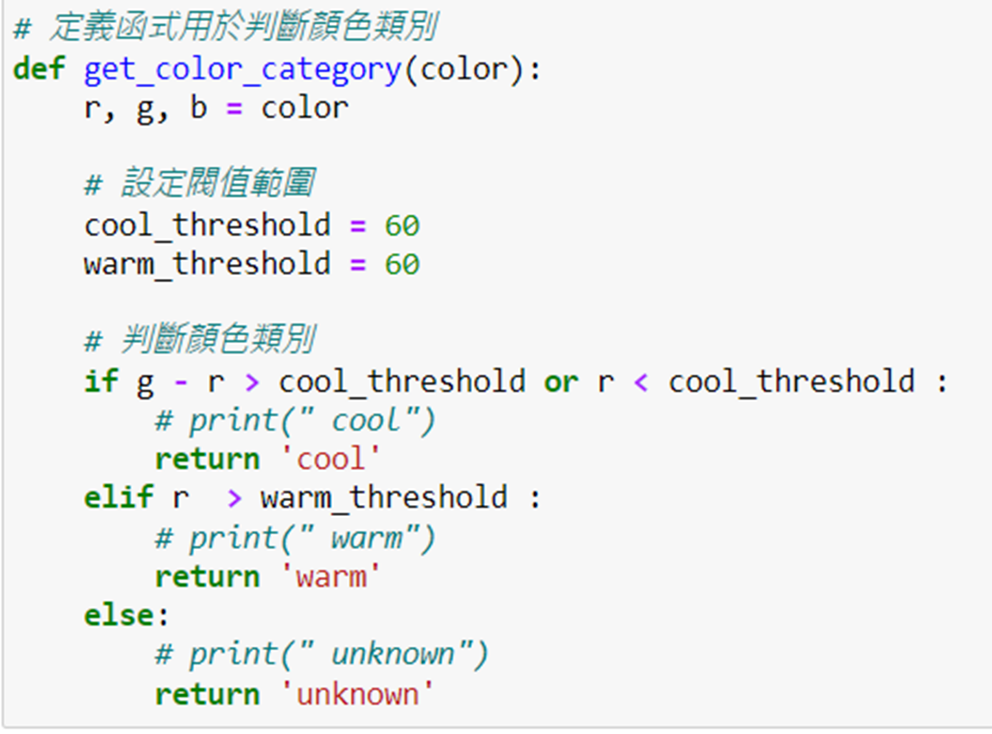
說明:

利用閾值 if…else…判斷屬於哪一個色系，且因為我的需求是想找出主要色系，所以乍看之下看不出是冷/暖的顏色，都會被歸類在unknown。

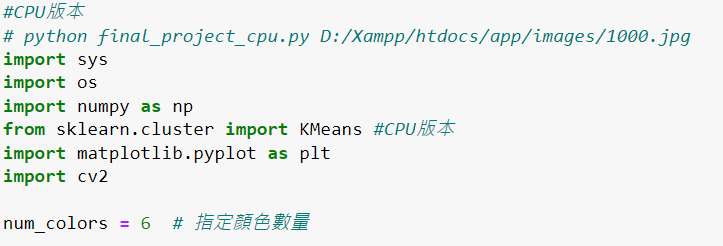
**步驟:**

* 輸入RGB值
* 判斷RGB值是否大/小於某個閾值。如果差異值大/小於閾值，則認為該區塊是暖/冷色。

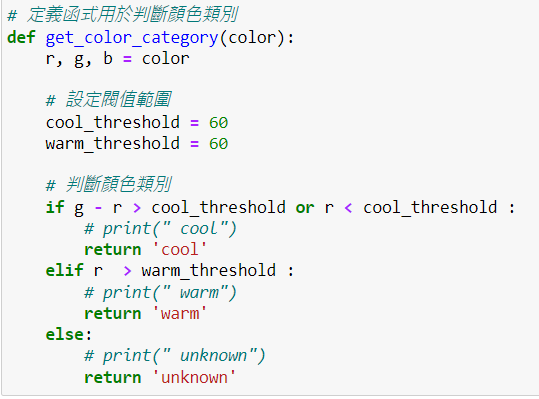
**程式碼:**



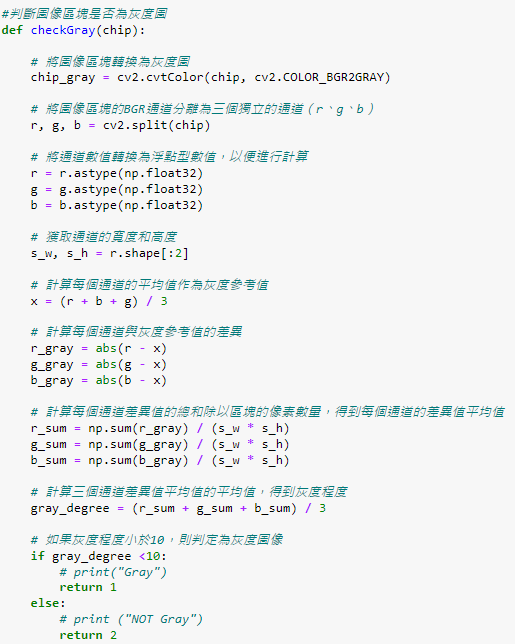
# 程式碼

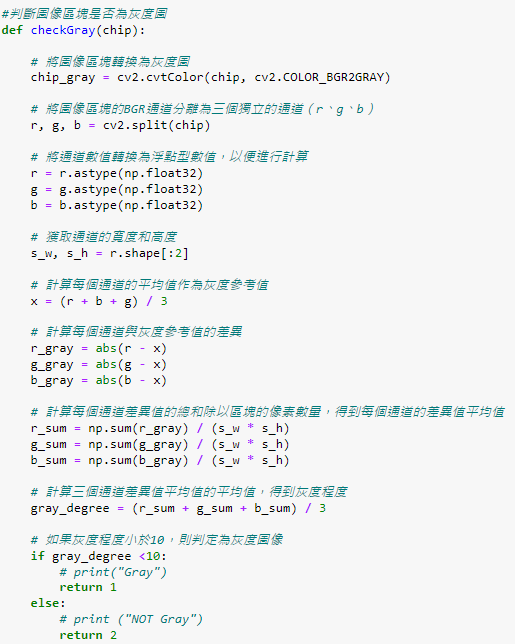


函式:判斷冷暖色



函式:判斷灰階

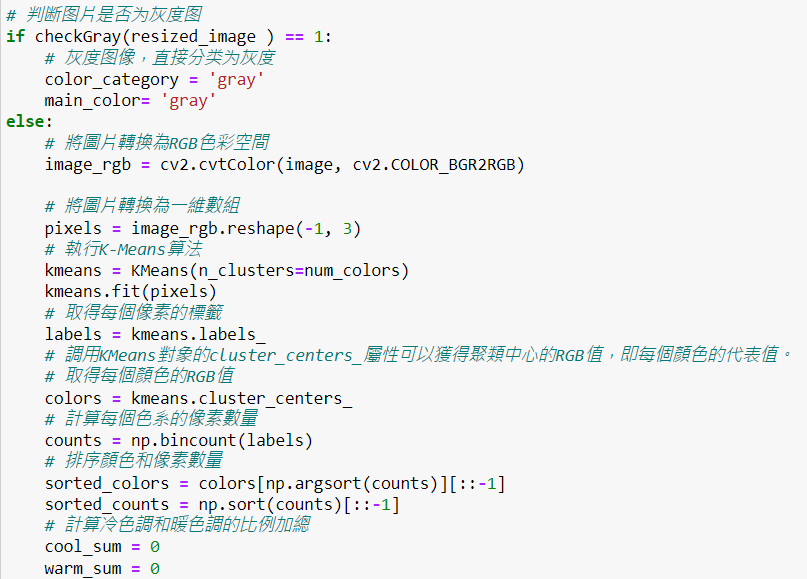


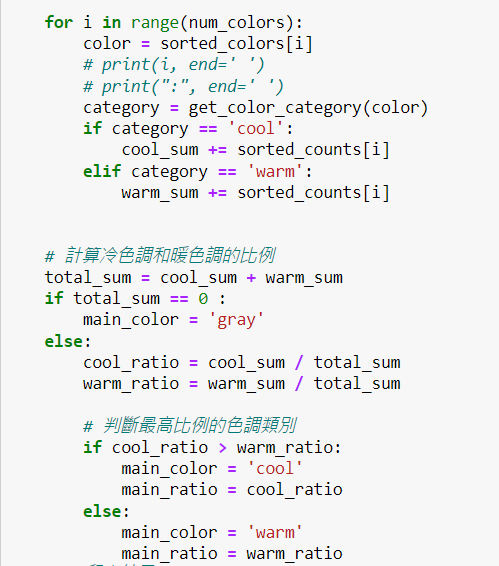


**主程式:**

由php 利用exec()函式 執行python



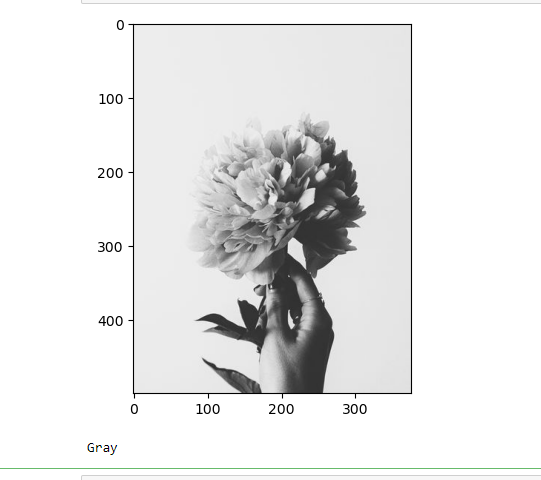






# 成果

* **灰階**



* **冷色系**

|  |  |
| --- | --- |
|  |  |
| **k-means 將照片分6群顏色，並判斷其於哪個色系 (分成warm cool gray三類)** | **根據各色系在圖中比例計算主要色系** |

* **暖色系**

|  |  |
| --- | --- |
|  |  |
| **k-means 將照片分6群顏色，並判斷其於哪個色系 (分成warm cool gray三類)** | **冷色系佔圖中8.4%，**暖色系佔圖中91.6%**，**可知暖色系為主要顏色 |

|  |  |
| --- | --- |
| **APP介面** | |
|  | |
| 資料庫內容 | |
|  |  |
| **照片上傳至server** | **辨識後從資料庫取得結果** |

* **結合APP介面實作成果**

按下上傳後資料庫新增a1001.jpg此筆資料，再經果AI辨識將顏色結果update到資料庫

# 總結心得

因為想要嘗試將APP和人工智慧結合，於是決定嘗試新題目，從一開始想分類底片，因為不同底片有不同色調，後來發現實作上有點困難，於是想到IG排版色調統一，從這個方向著手，因為是從0開始，也沒有人做過類似的題目，所以剛開始卡關很久，加上對Android Studio 還不夠熟悉，還有硬體設備因素，自己每天在圖書館做都還是沒甚麼進度，想法跟怎麼做都很清楚，但當機、debug就花了不少時間了，還好後來至少有做出大概的架構，細節防呆的部份需要在另外花時間完成。

這次是我第一次自己一個人開發一份專案，很有成就感，因為以前都是小組做專題，所以debug或是構想都可以一起討論，但這次所有事情都要自己來，變成依賴網路資料、Chat GPT，再將得到的資訊過濾，因為我發現Chat GPT回答不一定都是正確的。

這次專題我學到如何尋找題目，並將構想修飾轉換成可以實作的題目，學會如何下查詢指令，都對以後研究所如何找論文題目很有幫助，老師在過程中也抽空幫我不少忙，給我一些提點，很謝謝老師，也祝老師在彰師教書可以順順利利。