**遷移學習(transfer learning)**

卷積神經網路CNN是含有一個或多個卷積層的神經網路，目前常用於機器視覺應用領域。如果我們設計一個架構深的CNN模型，在經過大規模的影像資料集來訓練後，該CNN的神經層應該能提取多種視覺元素，我們可以應用於識別其他未知的影像。例如若模型能辨識凹型紋理，圓形物體與白色，則將這三種特徵經適當的方式結合，可以辨識出高爾夫球(當然若是其他顏色的高爾夫球就有可能誤判，故強調適當的特徵組合)。那要怎麼做呢?

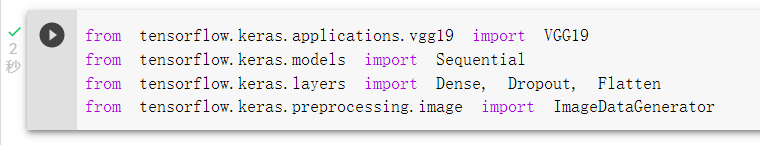
**遷移學習範例**

遷移學習(transfer learning)是一種技術，可以將以訓練過的機器學習模型移植到別的模型，無須從頭開始訓練新模型。

接下來，我們使用Cola開放平台(可參考文末說明)習此次的課題。

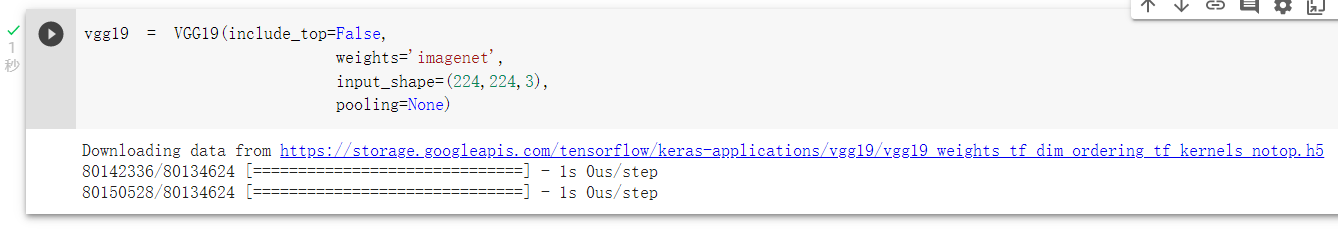
實作流程如下:

#### 步驟一， 匯入模組



載入tf.Keras預先過的模型(VGG19)。

步驟二，建立VGG19模型物件

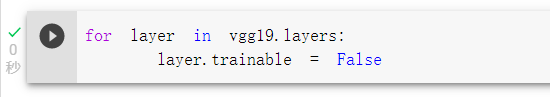
執行前須設定參數

1.include\_top=False:不包含終端的密集分類層。

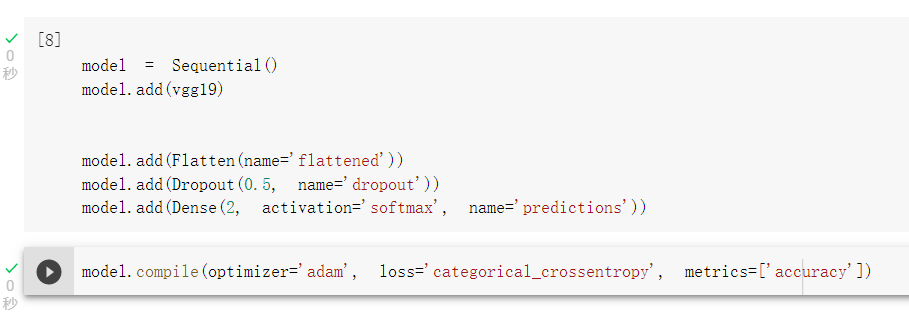
2.weights=’imagenet’設定載入的模型權重為使用ImageNet資料集訓練的結果。

3.input\_shape=(224,224,3)初始化模型的輸入影像形狀影像維度。

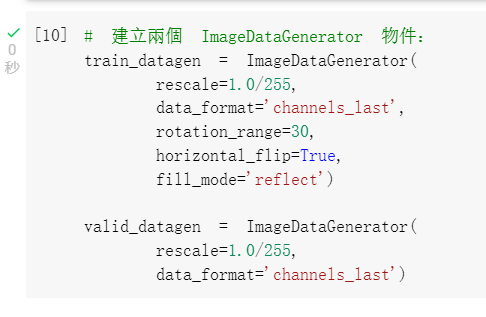
步驟三，凍結VGG19模型神經層



步驟四，添加新的神經層及執行編譯



步驟五，使用 ImageDataGenerator 類別



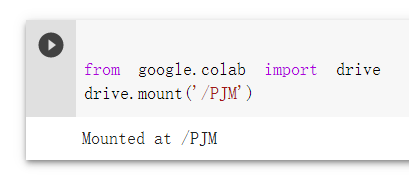
訓練資料生成器(train\_generator)會對影像執行下列作業

1.隨機執行旋轉(不超過30度)與水平翻轉。

2.正規化資料到0與1範圍內，即為除以255。

3.指定輸入資料形狀維224\*224\*3，顏色通道數出現在最後一軸data\_format=’channels\_last’。

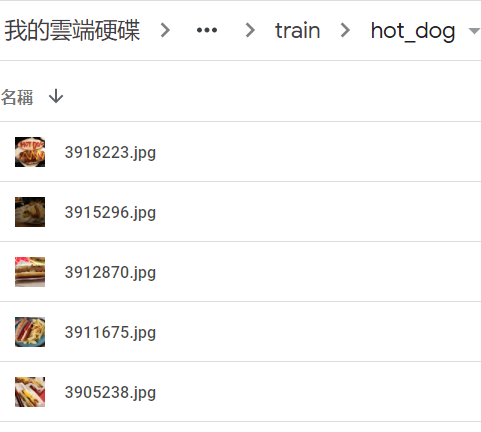
#### 步驟六，定義訓練與驗證資料的生成器



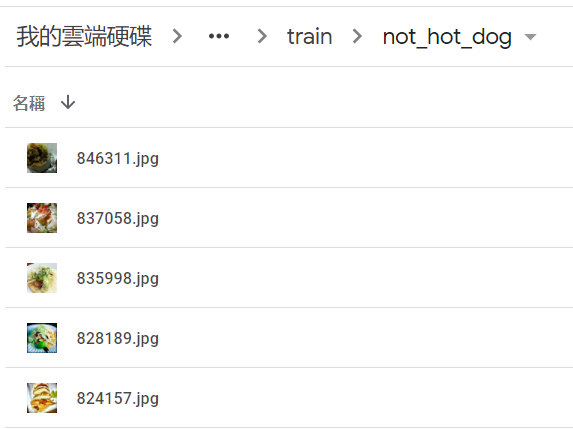
注意檔案目錄位置及在執行需要先mounted Google Driver，否則會沒有訓練資料。

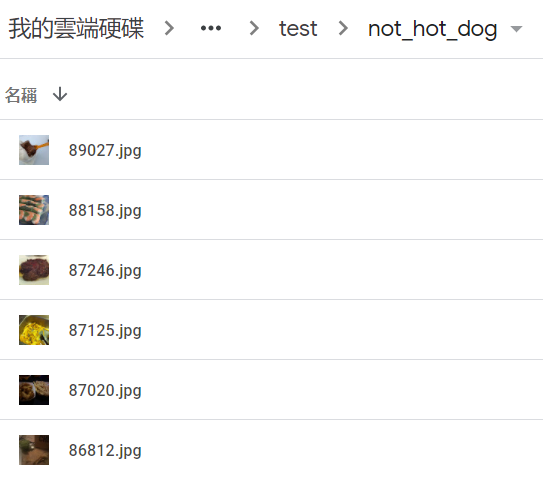


分別檢視Test & Train是否有資料

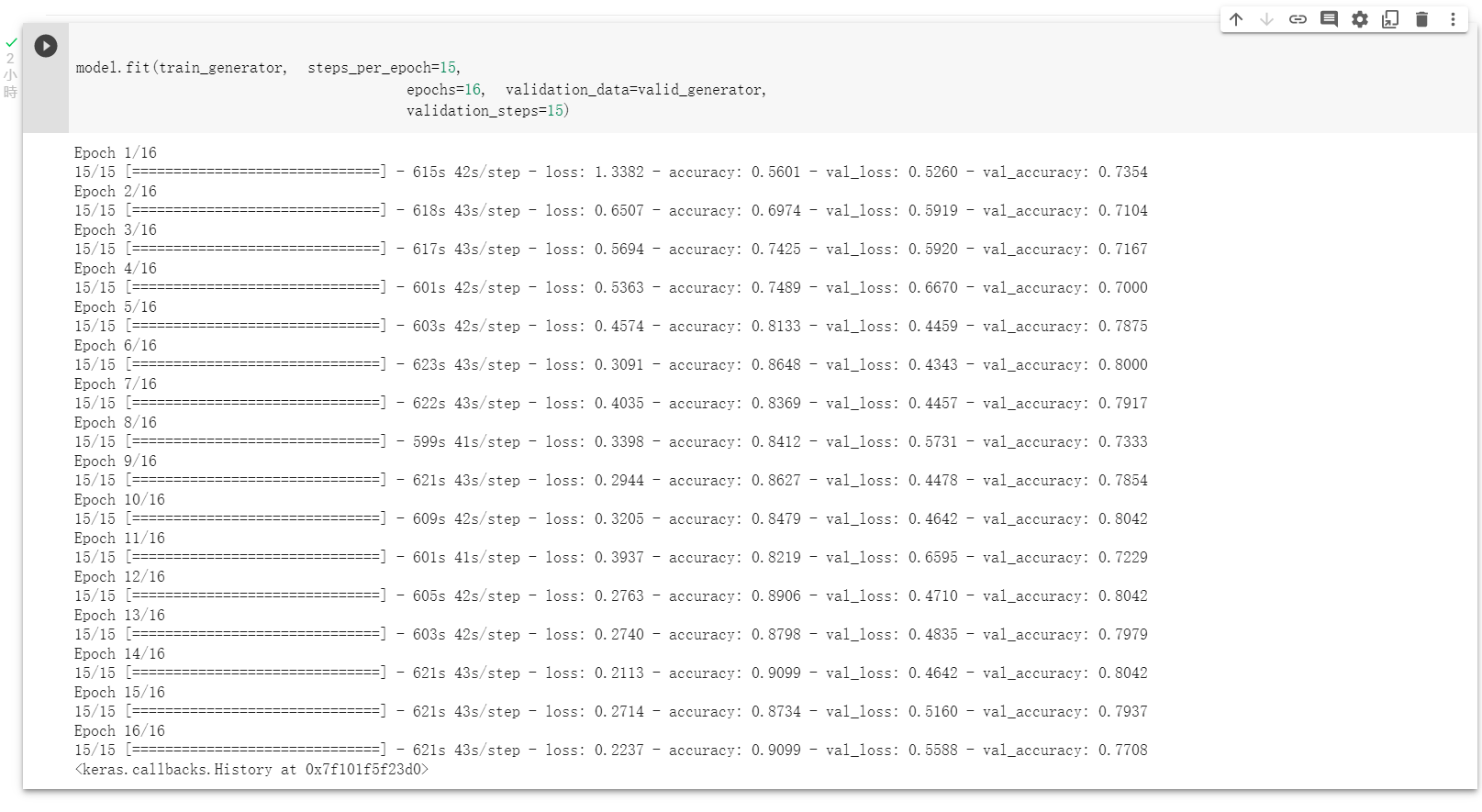








#### 步驟七，開始訓練

訓練結果如下:

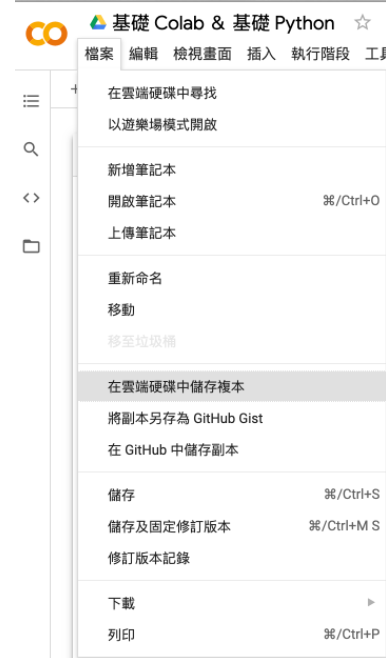
最佳準確率為0.8042，若是再花些時間調整參數，讓遺失率更低，辨識效果會更好。

**附錄:前置作業Colab環境準備**

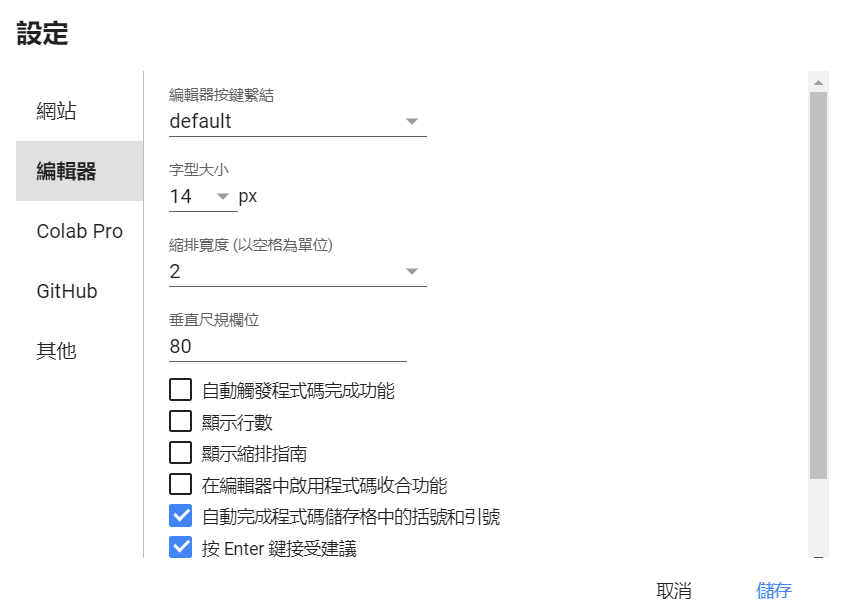
1.必須有自己的google帳號:使用自己的個人的 Google 帳號才可以執行自己 Colab上的程式。

2.先下載 Chrome瀏覽器，並登入 Google 帳號。

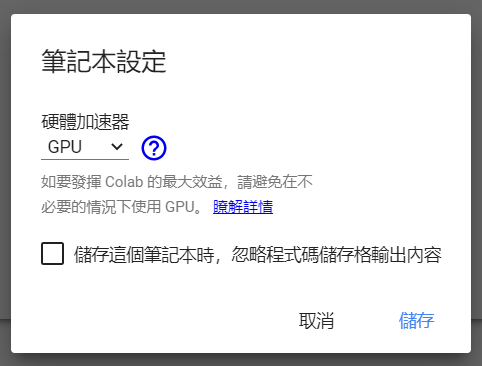
3.建立副本至個人目錄，否則不能存檔。



4.在⼯具 > 設定>編輯器中,反勾選「⾃動觸發程式碼完成功能」。



5.(必須做)設定執行階段類型。



6.療癒模式(非必須)，在⼯具 > 設定>其他, 勾選 「柯基⽝模式」和「貓咪 模式」，就有可愛的貓 貓狗狗來陪你寫程式了。



