# 中華民國第 60 屆中小學科學展覽會

# 作品說明書

---

國中組　生活與應用科學(一)科

## (鄉土)教材獎

032806

## 「大」「人」「物」洩天機

學校名稱：高雄市立五福國民中學

| 作者： | 指導老師： |
| --- | --- |
| 國一 陳艾妘 | 陳佳琪 |
| 國二 邱立凱 | 陳宗慶 |
| 國一 蔡鎔安 | |
| | |

關鍵詞：卷積神經網路 CNN、天氣預測、機器學習

# 摘要

天氣預測是攸關人類生活的大事。目前天氣預測主要利用大氣氣壓、溫濕度、雲層等參數資料來進行天氣預測。為達到低成本及小範圍之天氣預測，本研究利用物聯網的『眼睛』-樹莓派相機來觀測雲層，配合 Arduino 感測器蒐集到的氣象參數輔助，以預測 30 分鐘後降雨的機率。本研究採用卷積神經網路(Convolutional Neural Network, CNN)來進行模型訓練，首先將雲層圖片提取 RGB 特徵向量，配合氣象資料輸入模型以進行模型訓練，所訓練之模型可即時預測未來降雨的狀況。本研究進一步以精密度、召回率、F-measure、及混淆矩陣來驗證模型成效，準確度皆達 80%以上，證明本研究訓練之模型預測降雨的準確性及可行性。

# 壹、研究動機

人類的生活與天氣有著密不可分的關係，從日常生活到農業生產活動都受到不同的天氣變化而影響。為了知道未來的天氣，人們普遍都會利用氣象局的天氣預報來計劃活動。但是現有的天氣預測僅提供大區域範圍的普遍預測資料，尚未能針對小區域範圍提供即時和確切的天氣預報。台灣地形崎嶇變化很大，常常這個地方晴天，不遠的另一區卻突然一陣急雨，因此本研究希望提出一套以大數據、AI 人工智慧及配合物聯網蒐集資料的機器訓練模型方法，來建立一個可預測未來 30 分鐘小區域範圍的降雨預測模型，最後進一步驗證本天氣預測模型的準確度。

# 貳、研究目的

一、探討如何利用 Arduino 的各種感測器，蒐集溫度、濕度、大氣壓力、UV、以及是否降雨等資料。

二、探討如何使用樹莓派(Raspberry Pi) 相機模組拍攝蒐集雲層圖片 ，作為模型預測之主要輸入資料。

三、探討如何將大量的雲層圖片進行特徵值提取，以建立並訓練模型。

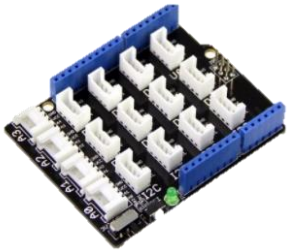
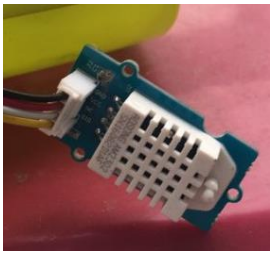
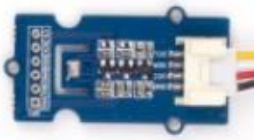
四、探討如何驗證訓練模型的準確度。

五、探討利用卷積神經網路不同隱藏層數量，對模型預測降雨準確度的影響。

六、建構並實際運用一套可即時預測降雨狀況的預測模型。

# 參、研究設備與器材

## 一、硬體設備:

| | | |
|---|---|---|
|  |  |  |
| 樹莓派 pi 3B | 樹莓派相機模組 | Arduino UNO |
|  |  |  |
| Arduino 擴充板 | 溫濕度感測器 | 大氣氣壓感測器 |
|  |  |  |
| UV 感測器 | 雨滴感測器 | 資料蒐集伺服器 |

## 二、軟體設備:

| 方法 | 環境/工具 | 說明 |
|---|---|---|
| 機器學習框架 | Keras | Python3 |
| 程式開發環境 | Google Colaboratory | GPU tesla T4 |
| OS | Google 線上環境 | Ubuntu 18.04 |

# 肆、研究過程與方法

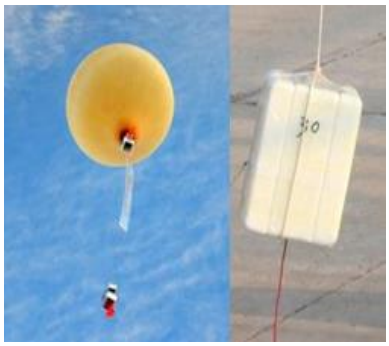
## 一、天氣預測現狀分析與瓶頸

### （一）天氣預測現狀

天氣預測是使用現代科學技術，對未來某一地點之天氣狀況進行預測。但因目前科技尚未最終透徹地了解大氣過程中複雜的物理現象，因此天氣預報與實際狀況總是有一定的誤差。由交通部中央氣象局所發布的一周天氣預測資料， 如圖一所示。

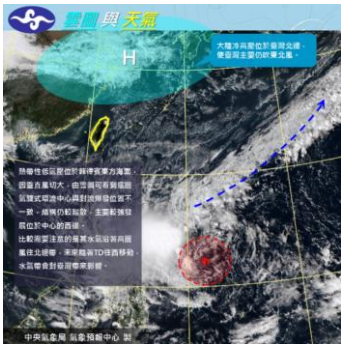

圖一:交通部中央氣象局一周天氣預測

1. 目前的天氣預測方式，包括地面的氣象站、高空氣球和所攜帶的探測儀(圖二)進行資料蒐集，再利用數學模型或者氣象方程式去計算出未來氣象的變化。
2. 地面的氣象站較為普遍，但其建置時須考慮各種條件，像是地形或者天災情況等。
3. 使用高空氣球可取得較精確的資料，但因高空氣球一直不斷的往上升最後會爆裂，而掛在氣球末端的無線電探測儀器，就會掉落損壞，屬於消耗性的儀器，因此其成本較高，如圖二。
4. 使用雷達以及衛星能蒐集到的資訊相對比較多，雷達會發射電磁波接收空氣中水滴的反射回波，進而計算水滴的大小或者相對位置，如圖三所示；而衛星可以捕捉雲層放出的紅外光以計算雲層的高度、溫度甚至是雲層的走勢，如圖四所示。但是這些設備還是有一定的限制，如雷達會受到地形的影響而掃描範圍變小。



圖二:高空氣球與探測儀

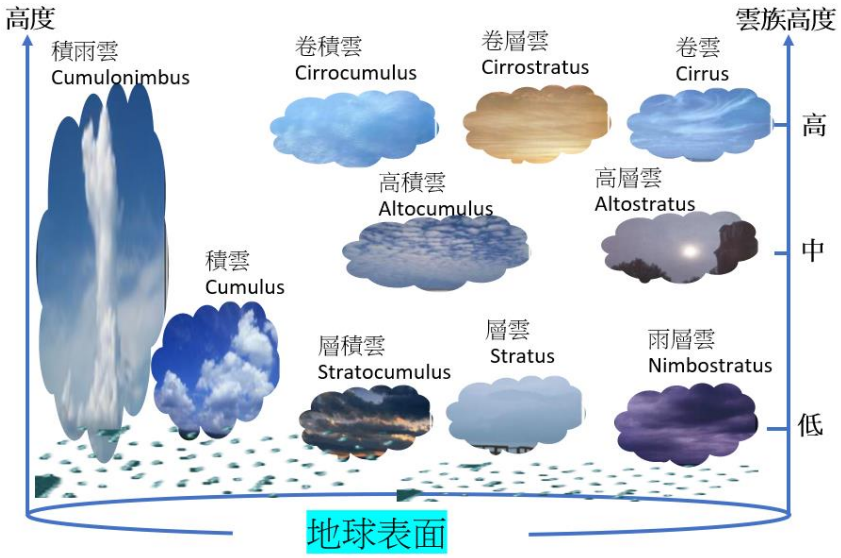
圖三:地面氣象站 L 波段雷達



圖四:衛星雲層觀測

（二）雲的分類與降雨

世界氣象組織(WMO)所出版的國際雲圖，依雲的高度及外形分成四屬十族，如圖五。不同的雲屬會伴隨著不同的天氣狀況。高雲族的雲即使產生降水，在下降過程中會蒸發而不致於落到地面。在低層次的雲屬中，雨層雲與積雨雲會帶來大雨。雲的基本辨識方法，可由高度、厚度、形狀、天氣狀況著手，但由於雲屬間存在著相互轉變的可能性，所以必須有賴一再觀察所累積的經驗，才能得到更準確的識別。因此為了提供一個更適用於一般人觀測雲層與降雨的方法，本研究計畫利用低成本的物聯網裝置，以及透過雲層影像與機器學習的相互輔助，來產生天氣預測的模型。




圖五: 雲層的分類

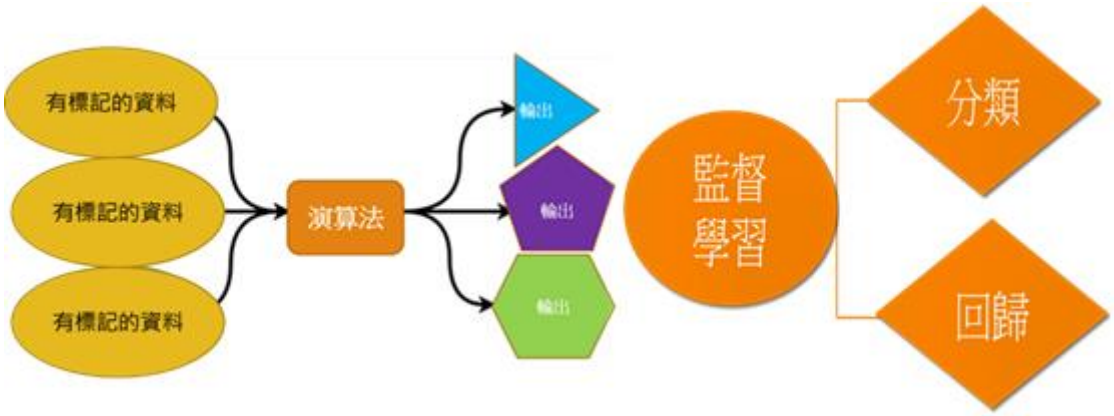
## 二、機器學習介紹

機器學習一詞於 1959 年由在電腦遊戲與人工智慧的先驅者 Arthur Samuel 提出，其概念主要是讓機器透過『學習』的方式，得到『推理』的能力，而機器學習也是人工智慧的分支之一。機器學習分為監督式學習、非監督式學習、強化學習以及深度學習，其架構如圖六所示。


圖六:機器學習架構圖

（一）監督式學習

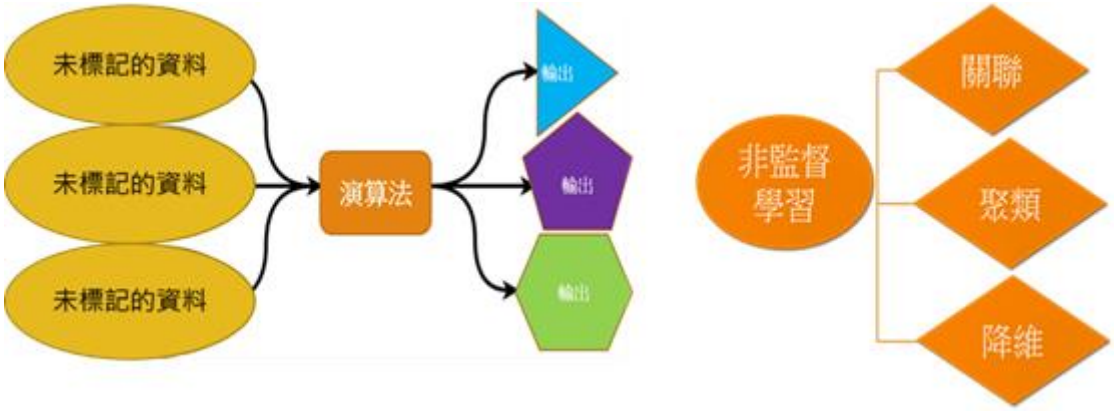
1. 在監督式學習中會利用已經標籤(Labeled) 好的資料進行訓練。
2. 標籤的資料就意味著這組資料有著正確的答案，機器在訓練的過程中會根據標籤判定預測正確或非正確的結果，藉由其結果更新模型的權重。圖七為監督式學習示意圖。




圖七:監督式學習示意圖

（二）非監督式學習

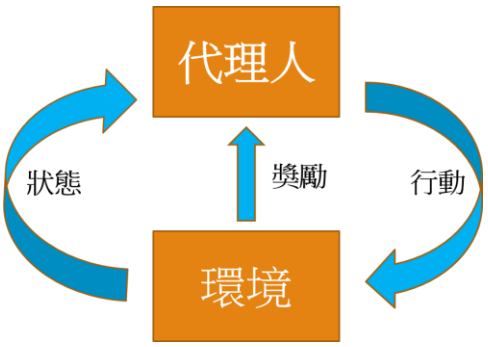
1. 在非監督式學習中訓練資料沒有任何的標籤，而機器會根據資料之間的差異學習出自己的判斷標準。
2. 一般非監督式學習主要應用在分群或分類上，圖八為非監督式學習示意圖。




圖八:非監督式學習示意圖

（三）強化學習

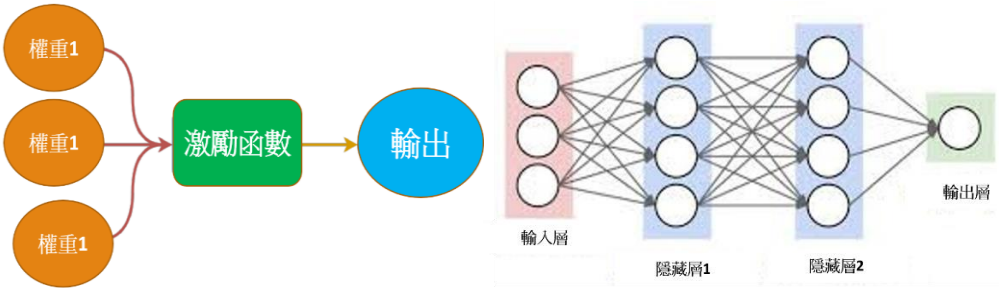
1. 強化學習中訓練資料沒有任何的標籤，甚至連輸入的訓練資料都沒有。
2. 強化學習以獎勵(Reward) 機制或者懲罰(Penalty) 機制讓模型訓練過程可以得知其行為是否有著更高的結果，最出名的代表就是 AlphaGo，圖九為強化學習示意圖。



圖九:強化學習示意圖

（四）深度學習

1. 深度學習（深度神經網路）是讓電腦可以自行分析資料找出「特徵值」，而不是由人類來決定特徵值，就好像電腦可以有「深度」的「學習」一樣。
2. 深度學習不但使用多層神經網路，同時使用「自動編碼器」（Auto-encoder）來進行「非監督式學習」（Un-supervised learning）。
3. 深度學習之多層結構由三個部分所組成，即輸入層、隱藏層以及輸出層，如圖十所示。輸入層主要是接收輸入的訊息；隱藏層為輸入層與輸出層的中間層，其層數及節點數皆不固定，最後輸出層為輸入資訊在中間傳輸過程中的最終輸出結果。



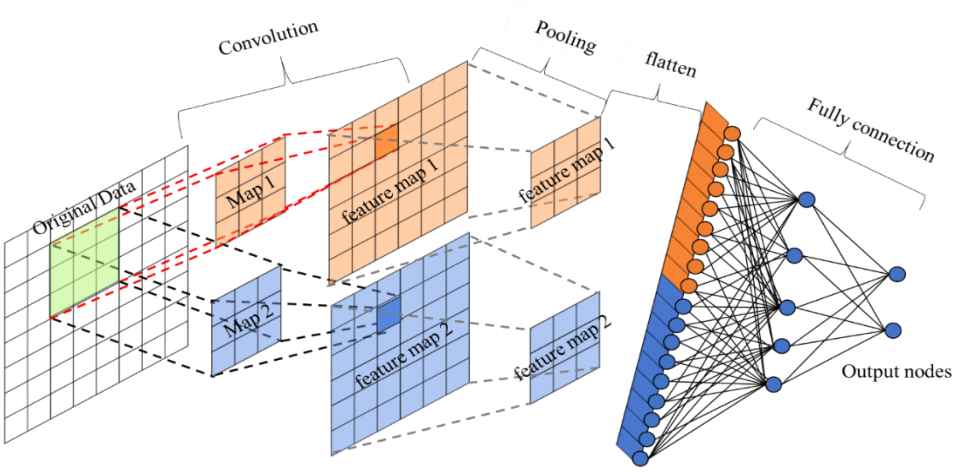
圖十:深度學習網路架構與示意圖

（五）多層感知機(Multilayer Perceptron, MLP)

多層感知機是一種類神經網路，屬於深度學習的一種。MLP 包含輸入層、隱藏層和輸出層，當資料傳輸至輸入層，經由一個或多個隱藏層之處理，最後在輸出層產生預測結果。目前 MLP 主要應用於分類預測問題。

（六）卷積神經網路(Convolutional Neural Network, CNN)

1. 卷積神經網路(CNN) 是一種前饋神經網路(Feedforward Neural Network)，在圖片辨識或者聲音辨識上都有著非常出色的表現。本研究便是利用其圖片辨識能力，應用於雲層圖片辨識。
2. CNN 是由多個卷積層(Convolution Layer)，池化層(Pooling Layer)，全連結層(Fully Connected Layer)、線性整流元(Rectified Linear Unit, ReLU) 等元件組成。圖十一為卷積神經網路的示意圖。




圖十一:卷積神經網路的示意圖

（七） 分類模型驗證

1. 一個訓練好的模型沒有經過任何的驗證評估，基本上不會承認其成效，又或者模型的成效不佳但是又不知道如何下手的時候，模型驗證的方法就派上用場了。
2. 混淆矩陣(Confusion Matrix) 、準確度(Accuracy)、精密度(Precision) 等都是一般驗證分類問題的方法。
3. 混淆矩陣(Confusion Matrix)：混淆矩陣是用於檢視機器學習分類的一種工具方法，特別是在監督式學習上，通常會以表一來呈現。一般混淆矩陣分爲真實類別(True Class) 也就是正確的標籤及預測類別(Predicted Class) 即預測的標籤。這兩類標籤分別都有正樣本(Positive) 以及負樣本(Negative)。

表一: 混淆矩陣

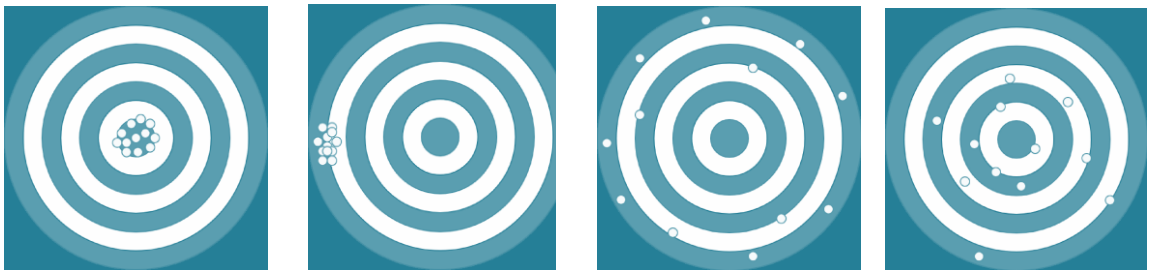
| | | 真實類別(True class) | |
|---|---|---|---|
| | | 正樣本(Positive) | 負樣本(Negative) |
| 預測類別 (Predicted class) | 正樣本(Positive) | True Positive(TP) | False Positive(FP) |
| | 負樣本(Negative) | False Negative(FN) | True Negative(TN) |

4. 準確度(Accuracy) : 準確度主要是衡量模型效能的一種參數之一，主要是根據預測的結果與真實結果做對比，如果對比之後的誤差越小則準確度較高，如果誤差越大準確度越低。準確度計算方式如公式(1)。

$$準確度=\frac{TP+TN}{TP+FP+FN+TN} \quad 公式(1)$$

5. 精密度(Precision)：精密度是指在預測類別中為正樣本，而真實類別也是正樣本的正確的比例。公式(2) 為精密度的計算方式，圖十二為各個高低準確度與高低精密度的示意圖。

$$精密度=\frac{TP}{TP+FP} \quad 公式(2)$$



(a)高準確度高精密度 (b)高準確度低精密度 (c)低準確度高精密度 (d)低準確度低精密度

圖十二: 高低準確度與高低精密度示意圖

6. 召回率(Recall)：召回率也稱之為敏感性(Sensitivity)，主要是計算真實類別之正樣本被成功預測出來的機率。其公式如公式(3) 所示，計算結果越高越好。

$$召回率=\frac{TP}{TP+FN} \qquad 公式(3)$$

7. F-measure：主要是計算精密度以及召回率的調和平均數公式如(4)所示。

$$F\text{-}measure=\frac{2\times Precision\times Recall}{Precision+Recall} \qquad 公式(4)$$

（八）Google Colaboratory 又簡稱為 Google Colab，是 Google 提供的雲端開發環境，主要作為 Python、機器學習和深度學習的輔助工具，有提供免費的 GPU 運算系統。使用者可直接利用 Google Colab 撰寫程式，並利用免費 GPU 訓練所建立之學習模型。本研究雲層圖片之模型訓練，就是在此平台測試完成。

## 三、研究架構概述：小型氣象站系統與機器學習分析架構

現代的天氣預測方法，幾乎都利用到超級電腦，但是超級電腦成本相當高，可能甚至連國家都不一定會擁有，而且演算的次數及條件也有相當的要求。因此本模型的設計方向包含：

（一）嘗試提出一套研究架構，利用機器學習的方法進行模型的訓練，訓練設備雖然運算能力較低，但成本相當低且容易上手。

（二）用 Arduino 的各種感測器，來蒐集溫濕度、大氣壓力、UV 和是否降雨等大氣資料。

（三）利用樹莓派的相機模組，拍攝蒐集天空的雲層圖片 。

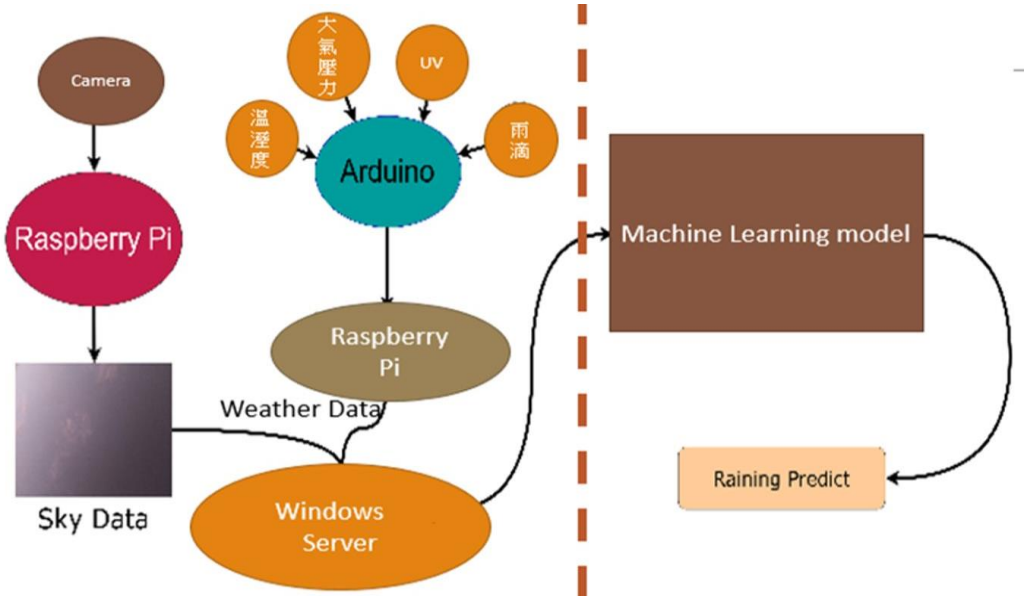
（四）通過樹莓派傳送到電腦伺服器。

（五）對雲層資料做初步的處理、特徵提取與增量。

（六）採用卷積神經網路進行機器學習來訓練模型，以建構 30 分鐘後是否降雨的預測模型。

（七）在模型訓練完成後，模型預測時不需再耗費大量成本，即可在很短的時間內對未來 30 分鐘的天氣做預測。

（八）雲層圖像與天氣資訊蒐集、處理以及機器學習模型訓練之架構簡示如圖十三。



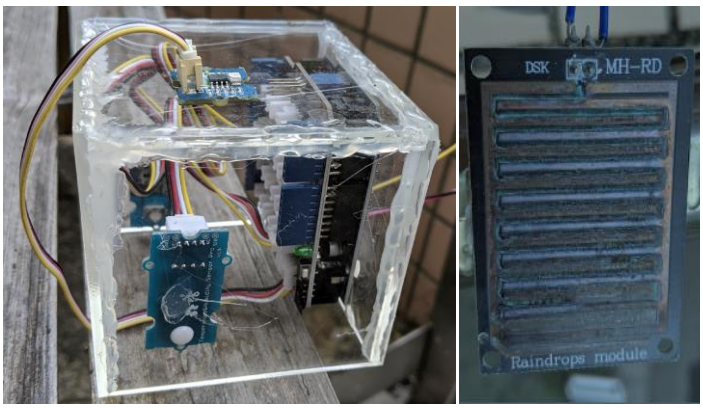

圖十三: 小型氣象站系統與機器學習分析架構

## 四、探討如何利用 Arduino 各式感測器，建立小型的氣象資料蒐集站

（一）說明

Arduino 是一種利用單晶片開發出來的介面控制板，可以結合各種線路、輸入和輸出裝置或各類感測模組，例如按鈕、LED 燈、液晶螢幕、超音波感測器、馬達等，製作各式各樣的儀器裝置或是有趣的玩具，例如門窗開關偵測警報器、自製創意樂器、自動射紙飛機玩具或各種功能的感應器等。

因其相對低成本特色，於是本研究利用 Arduino 結合各種感測器模組，來建構氣象資料蒐集站，考量設備的放置環境與傳統氣候預測參考的變因，本研究搜集的大氣資料包含溫度(Temperature)、濕度(Humidity)、大氣壓力(Atmospheric Pressure)、UV(UltraViolet, UV)及是否降雨等。這就是利用「物」聯網（Internet of Things，簡稱 IoT）的概念，達到連結實體物件與虛擬數據來蒐集資料。本設計裝置如圖十四所示。




圖十四: Arduino 以及感測器元件

（二）實驗步驟

1. 組裝溫、濕度、大氣壓力、ＵＶ與雨滴感測器模組。
2. 編寫 Arduino 感測器之程式碼，設定感測器為每 30 秒感測一次。
3. 感測資料透過樹莓派與 putty 程式連結，自動將蒐集資料傳輸至電腦伺服器。
4. 每日確認設定無誤並整理蒐集的數據。
   （1） 設定 Arduino ip: 192.168.1.3
   （2） 設定 hostname: raspberrypi.local
   （3） 伺服器 IP 位置確認指令: ifconfig; ping 192.168.1.9
   （4） 時間設定確認指令: date; sudo date --s='2019-09-15 10:33:00'
   （5） 確認目前工作流程 ps aux l grep python; kill process id
   （6） 程式碼執行指令: sudo python3 run.py
5. 本研究採用的電子元件整理如表二。

表二: 實驗感測器介紹

| 設備名稱 | 設備型號 | 功能 |
|---|---|---|
| 樹莓派 Pi 3 | 3B | 將圖像與感測器資料傳回伺服器 |
| 相機模組 | Noir module v2 | 拍攝雲層圖片 |
| Arduino | UNO | 將感測器資料傳回給樹莓派 |
| Arduino 擴充版 | Base shield Grove | 擴充感測器插槽 |
| 溫溼度感測器 | Grove Temperature Humidity sensor pro | 測量溫度與濕度 |
| 大氣壓力感測器 | Grove Barometer | 測量大氣壓力 |
| UV 感測器 | Grove UV sensor | 測量 UV 強度 |
| 雨滴感測器 | 水滴型雨水感測器 | 測量是否下雨 |

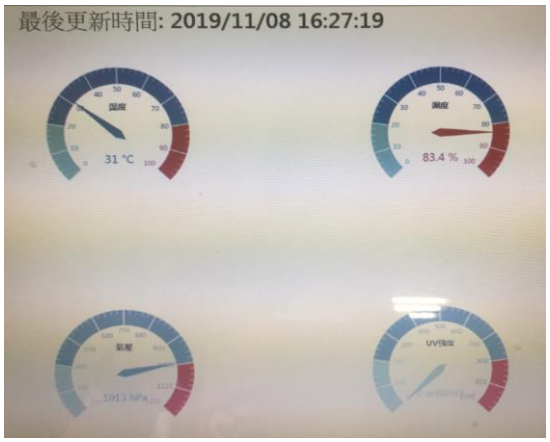
（三）實驗結果

1. 透過安裝好的小型氣象站，我們蒐集了相當多筆的大氣資料，舉例 10 月 13 日的天氣數據整理如表三 。另外，我們也利用網頁即時呈現擷取的氣象資料如圖十五。
2. 由此圖可以發現，即使大氣資料相同，降雨狀況也可能不同，所以傳統的大氣資料其實對預測降雨無法達到很好的解釋，這也是本研究希望透過機器學習來建構一套更有效的模型預測的動機。
3. 收集的大氣資料，用在機器模型的訓練資料中，最重要的是把“下雨狀態”當作 CNN 模型裡的標籤欄位，其方式是以雨滴感測器感測到的數值(介於 0 至 1023)做標記，當數值為 0 至 341 標記為“不會降雨”，342 至 682 標記為“ 可能會降雨”，683 至 1023 標記為“會降雨”。 在之後進行模型預測時，預測結果會對應到 30 分

鐘後是否降雨，來判斷是否預測成功，其容忍值為±5 分鐘。

表三: 10 月 13 日的天氣數據(範例資料)

| 時間<br>戳記 | 大氣壓力<br>(hPa) | 溫度<br>(˚C) | 濕度<br>(%) | 紫外線<br>(mW/m$^2$) | 下雨狀態<br>(Raining) |
|---|---|---|---|---|---|
| 2019/10/13 | 1006 | 26.2 | 94.4 | 0 | Rain Warning |
| 2019/10/13 | 1006 | 26.2 | 94.425 | 0 | Rain Warning |
| 2019/10/13 | 1006 | 26.2 | 94.5 | 0 | Raining |
| 2019/10/13 | 1006 | 26.2 | 94.5 | 0 | Rain Warning |
| 2019/10/13 | 1006 | 26.2 | 94.475 | 0 | Rain Warning |
| 2019/10/13 | 1005 | 26.5 | 93.6 | 18 | Not Raining |
| 2019/10/13 | 1005 | 26.5 | 93.48 | 22.4 | Not Raining |
| 2019/10/13 | 1005 | 26.5 | 93.35 | 37.25 | Not Raining |
| 2019/10/13 | 1005 | 26.52 | 93.3 | 51 | Not Raining |




圖十五:即時呈現擷取到的感測器資料畫面

## 五、探討如何使用樹莓派(Raspberry Pi) 相機模組作為雲層拍攝的裝置，收集雲層圖片作為預測資料

（一）說明

樹莓派是一種微型電腦，可以連接各種輸入、輸出裝置，例如：螢幕、滑鼠、相機等多種配件，在樹莓派上就可以編寫程式控制裝置，讓相機定時拍攝，或縮時攝影等各式各樣的功能。本研究選擇了樹莓派作為雲層拍攝的裝置，是因成本較低且建置容易，與使用衛星觀測相較之下簡易可行許多。且採用從地面往天空拍攝的方式，與衛星是由天空往地面拍攝也不同。

（二）實驗步驟

1. 組裝樹莓派相機模組如圖十六，設定以 45°角左右向上進行拍攝。
2. 編寫相機拍攝程式碼，設定每分鐘進行一次拍攝。
3. 所拍攝的照片如圖十七，透過 putty 程式自動將蒐集的圖片資料傳輸至電腦伺服器。
4. 每日確認設定無誤並整理蒐集的數據。
   (1) 設定樹莓派相機模組 ip：192.168.1.108
   (2) 設定樹莓派相機模組 hostname: camera.local
   (3) 伺服器 IP 位置確認指令： ifconfig; ping 192.168.1.9
   (4) 時間設定確認指令：date; sudo date --s='2019-09-15 10:33:00'
   (5) 確認目前工作流程 ps aux l grep python; kill process id
   (6) 程式碼執行指令: sudo python3 run.py

圖十六: 樹莓派與相機模組


圖十七:即時拍攝到的雲層畫面

（三）實驗結果

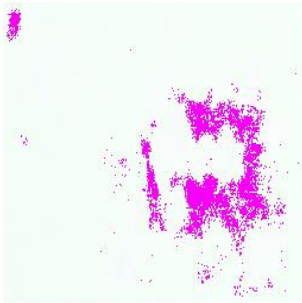
1. 因夜晚的圖片清晰度不夠，加上一些特徵可能因低光源下被忽視、雜訊會比白天來的高或者曝光不足，而無法擷取需要的資訊等，所以爲了簡化問題，本研究暫不考慮夜晚的情況。
2. 雲層圖片拍攝後，會先調整尺寸以利電腦運算，並初步進行提取特徵，轉換為特徵圖，如圖十八所示。


圖十八:天氣雲層經轉換後的圖片

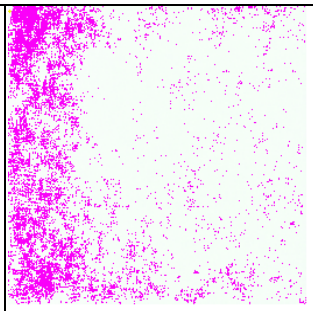
3. 拍攝的雲層圖片會配合大氣資訊的降雨標籤，區分成不會降雨（Not Rainging）、可能會降雨(Rain Warning)與會降雨(Raining)等三類，蒐集的資料舉例如表四。
4. 雲層圖片及氣象資料是由 7 月到 10 月蒐集，其中因 9 月機器損毀，大部分為非正常的情況。雲層圖片總共有 25113 筆資料，其中 11029 筆資料為無效資料(判定為夜晚的圖片或無法辨識)，14084 筆為有效資料；氣象感測器總共有 182181 筆資料，其中 11989 筆沒有降雨，2789 筆為接近降雨和 1536 筆為降雨，105867 筆資料為無效資料(異常值或者亂碼)，總共的資料如表五。

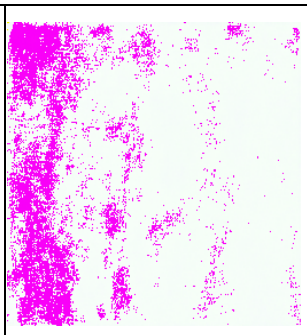
表四:天氣雲層照片蒐集舉例(左邊圖片為原始照片，右邊圖片為特徵圖)

| 時間 | 2019/10/11 12:00 |
|---|---|
| 大氣壓力 | 1016 |
| 溫度 | 29.65 |
| 濕度 | 59.1 |
| UV | 230.098 |
| 下雨狀況 | Not Raining |

| 時間戳記 | 2019/08/19 10:00 |
|---|---|
| 大氣壓力 | 1020 |
| 溫度 | 26.51 |
| 濕度 | 88.714 |
| UV | 56.324 |
| 下雨狀況 | Rain Warning |

| | | | |
|---|---|---|---|
| 時間 | 2019/08/16 10:00 |  |  |
| 大氣壓力 | 1017 | | |
| 溫度 | 27.248 | | |
| 濕度 | 85.294 | | |
| UV | 15.242 | | |
| 下雨狀況 | Raining | | |

表五:全部蒐集資料的雲層圖片和氣象資料統計

| 資料類型 | 有效資料/無效資料 | 說明 |
|---|---|---|
| 雲層圖片 | 14084/11029 | - |
| 氣象感測器資料 | 16314/105867 | 沒有降雨: 11989<br>接近降雨: 2789<br>降雨:1536 |
| 可用訓練資料 | 14084 | 沒有降雨: 11924<br>接近降雨:1535<br>降雨:625 |

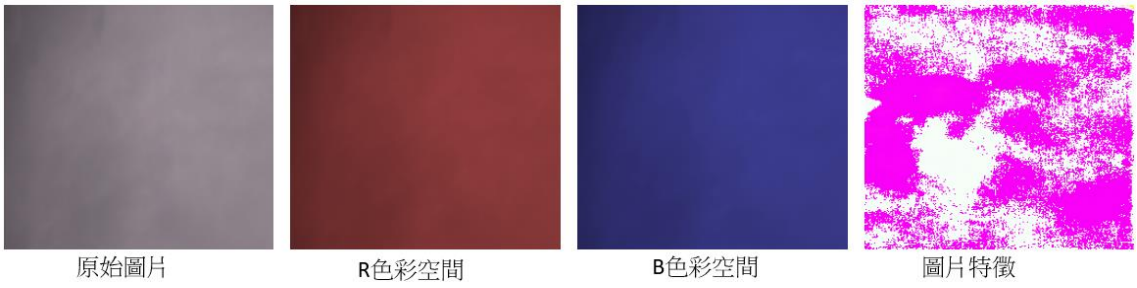
## 六、雲層圖片的特徵提取

（一）說明

1. 圖片是由一小格一小格的像素所組成，一張 256x256 解析度的圖片就是包含 256 列 x256 行的像素。灰階圖片就是黑白照片，明暗不同，可由一個數字來代表，數值 0～255，0 通常代表灰階中最暗的顏色，也就是黑色，255 通常代表灰階中最亮的顏色，也就是白色。彩色圖片則是由紅色（R）、綠色（G）、藍色（B）三種色階組成，每一種顏色也都由數值 0～255 來表示，例如（R,G,B）＝（0, 0, 255）就是藍色，舉例如圖十九。電腦就是利用這些數值來讀取圖片，而顯示對應的顏色，256x256 的彩色圖片電腦可解讀為 256x256x3 的數字矩陣，也就是三個與圖片大小相符的矩陣，也稱為R G B三個通道（channel）或深度（depth）。


圖十九: 圖片 RGB 通道數值舉例

2. 根據參考文獻可知，當雲層發生時，環境之原有色彩會遭到雲層的掩蓋而呈現灰階色彩。而灰階色彩具有一項重要的 RGB 特徵，那就是三原色的強度相等，舉例來說白色為(255,255,255)、灰色為(127,127,127)、黑色則是(0,0,0)。因此，我們利用 R、G、B 三色強度值之差值做為雲層檢驗的標準，當 RGB 之差值越小，其組成越接近灰階色彩，也就代表越有可能存在雲層。當差值介於 30 到-30 之間，判斷此像素點為雲的部分。故本研究在機器學習前，將先做雲層特徵的提取，以利機器學習，而提取雲的特徵較佳的方式，即是先進行 RGB 通道差值的運算，轉換原始圖片為 RGB 通道差值來保存，以做為 CNN 卷積神經網路的原始輸入資料，如圖二十。




圖二十:雲層圖片的特徵擷取

3. 而 CNN 的卷積層是將輸入圖片的 RGB 差值矩陣透過核心（kernel）進行卷積運算，經過矩陣運算後，可以得到新的圖片。其中核心可視為濾鏡（filter）或濾波器，目的是讓圖片特徵更加明顯，所以新的輸出圖片我們也稱為特徵圖（feature map)，當做下一層卷積層的輸入圖片，將特徵向下傳遞，如此一層一層的卷積神經網路，就是為了要達到圖片特徵提取的功能，簡化運算的複雜度。
4. 隨著科技進步，硬體計算能力提升，尤其 GPU 晶片的開發讓人工智慧機器學習的發展越來越迅速，但對於個人開發而言，這種高階運算的 GPU 還是相當貴，還好 Google 公司在其 Google Drive 的應用程式 Google Colaboratory（Colab）上，提供了免費的 GPU 讓想進行機器學習實驗的使用者進行運算，我們接下來的 CNN 模型即是在 Colab 的環境上執行。

（二）處理程序說明

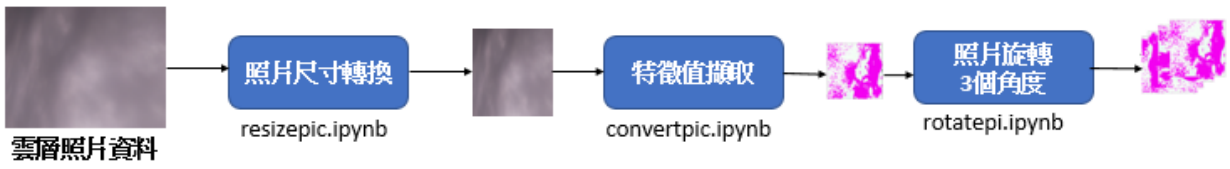
1. 結合雲層圖片資料與感測器的大氣資料，作為模型訓練的輸入資料。


2. 將雲層圖片大小做尺寸的轉換，由原來的圖片大小解析度為 1640x1232，轉換為 256x256。此步驟的目的是為了讓電腦可以進行讀取，同時又降低運算的複雜度。


3. 再利用新尺寸的圖片進行 RGB 通道差值的運算，取得初步的雲層特徵圖。


4. 最後為了提高準確度，我們考量在既有的雲層資料裡，增加可作為分析的雲層圖片的資料量。


5. 總結圖片處理程序流程會如圖二十一所呈現。




圖二十一:大氣感測器資料與雲層圖片的處理程序

（三）實驗結果

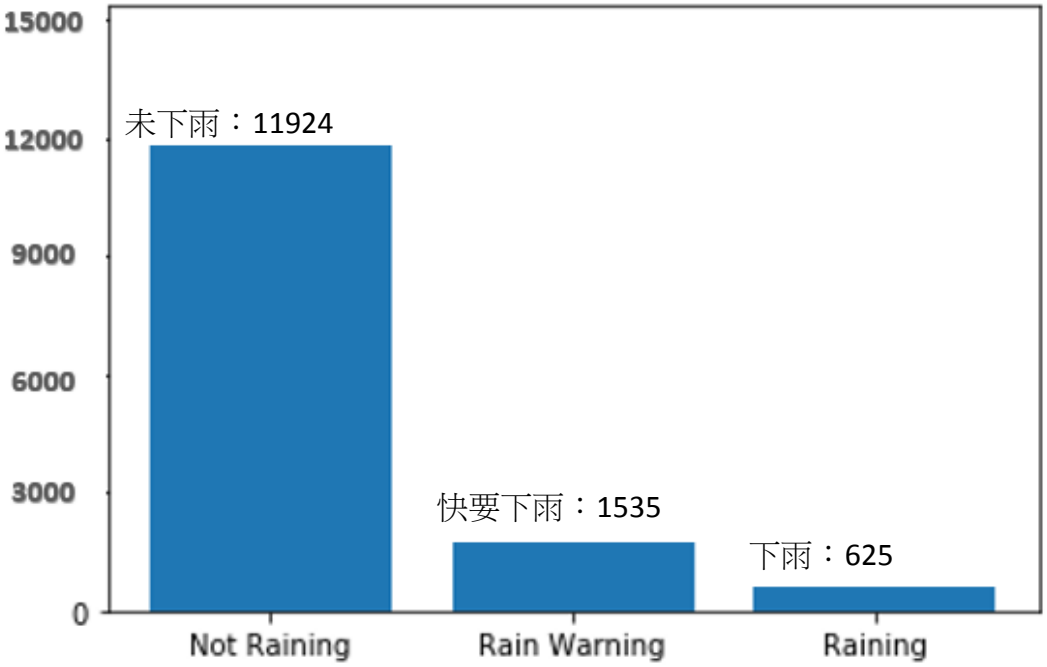
1.結合雲層圖片資料與感測器資料

（1） 利用 colab 開啟 data_combinate.ipynb

（2） 開啟圖片和大氣資料的檔案，將對應時間的資料做結合。

（3） 刪除無效的資料或沒有配對的資料。

（4） 整理後的資料如下圖二十二所示：




圖二十二: 雲層圖片對應降雨狀況統計圖

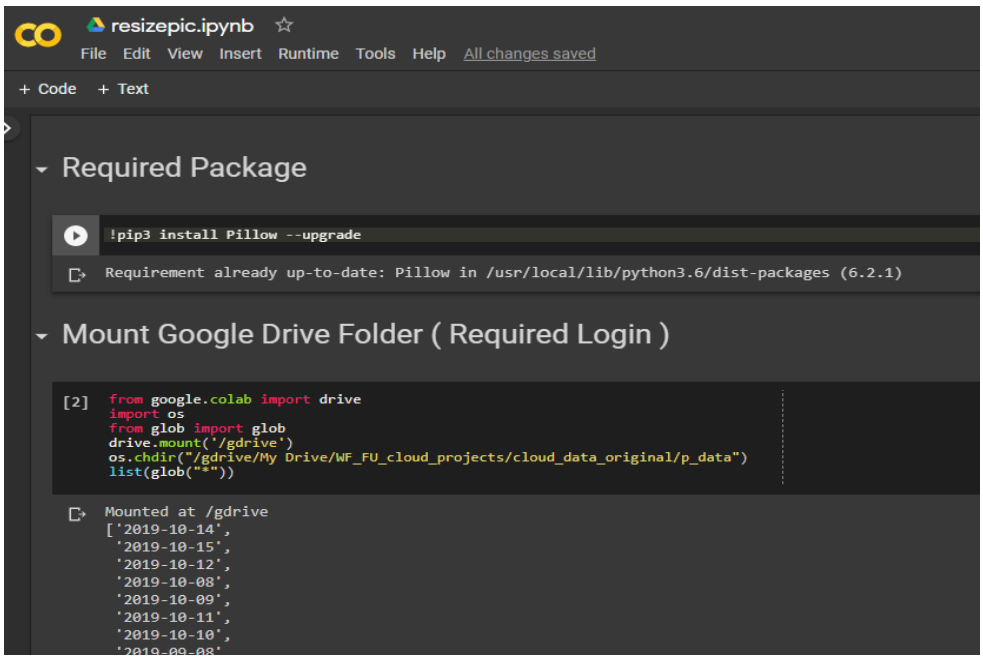
2.雲層圖片尺寸轉換

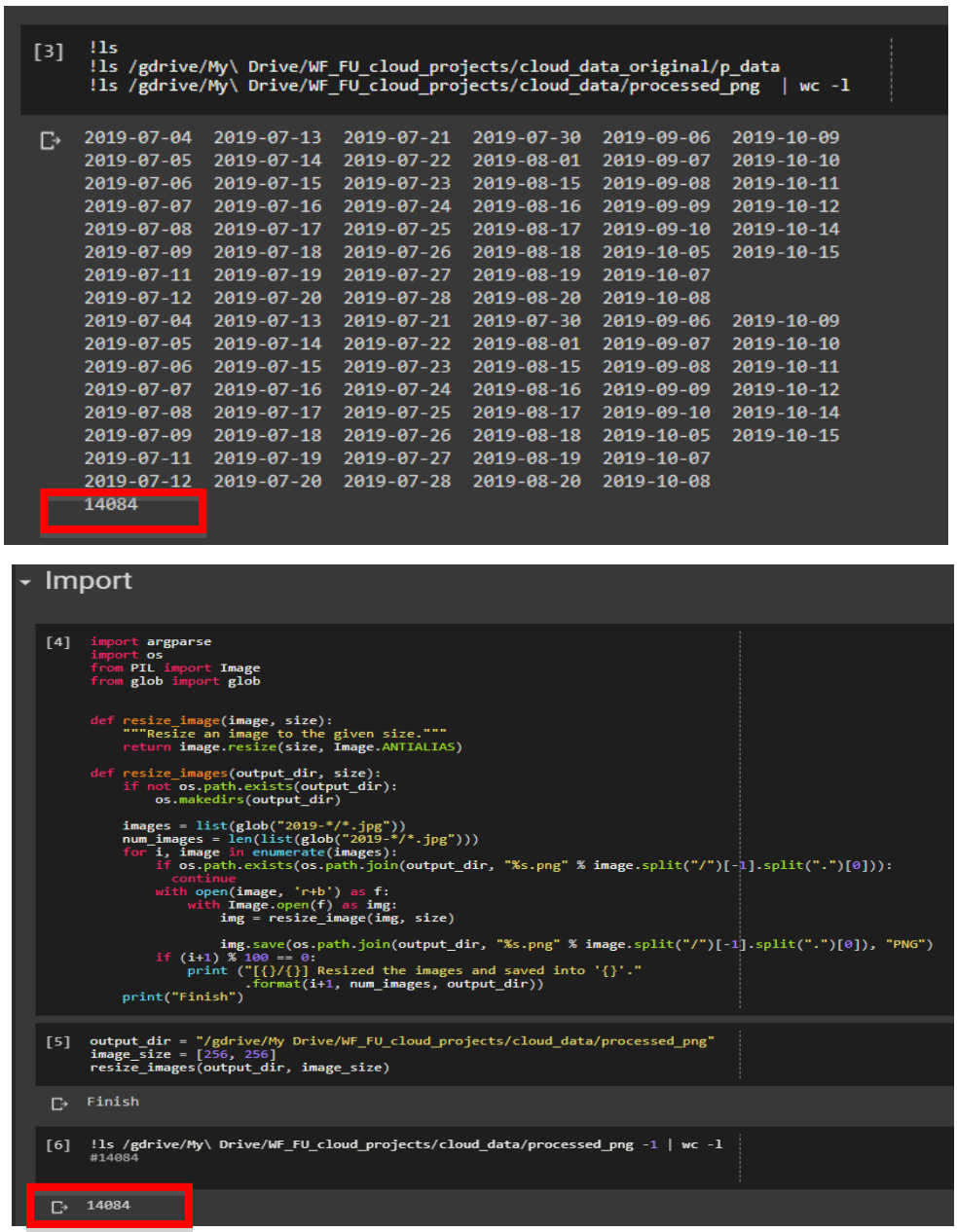
（1） 在 colab 上開啟程式 resizepic.ipynb，執行 Runtime(run all)，之後程式會照順序執行。

（2） 查看雲層圖片資料 (p_data)和 processed_png 目錄總共找到 14084 個雲層圖片檔。

（3） 完成 resize 的動作並輸出檔案至我的雲端硬碟 > Cloud_projects > cloud_data > processesd_png。

（4） 程式執行過程如下圖二十三。






圖二十三: 照片尺寸轉換程式執行程序示意

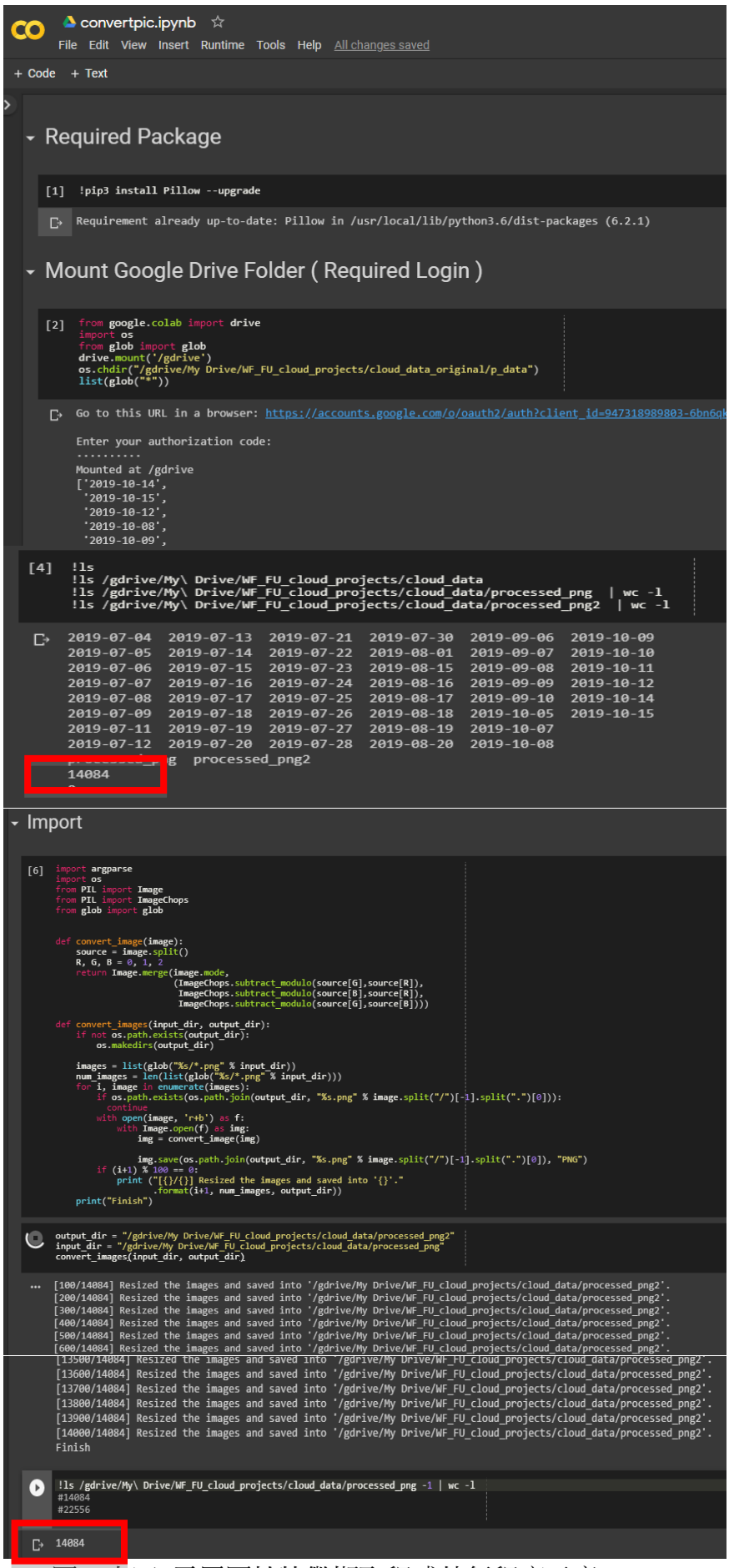
3.雲層圖片特徵擷取

（1） 用 colab 開啟 convertpic.ipynb 進行轉換以擷取雲層圖片特徵。

（2） 查看 processed_png 和 processed_png2 目錄皆有 14084 張圖片檔。

（3） 完成 convert 轉換特徵的動作並輸出檔案至我的雲端硬碟>Cloud_projects > cloud_data > processesd_png2。

（4） 程式執行如圖二十四。




圖二十四: 雲層圖片特徵擷取程式執行程序示意

4.旋轉圖片增加雲層圖片資料量

大數據是機器學習成功的重要關鍵，故為提升模型的準確度，在資料蒐集時間限制下，我們思考如何進一步增加特徵擷取過的雲層圖片資料，考量方向於是將雲層圖片做 90 度旋轉、180 度旋轉、270 度旋轉後的圖片納入分析，如此可以讓資料量增加為原有的四倍，雖然運算時間會增加，但可以符合大數據的概念。

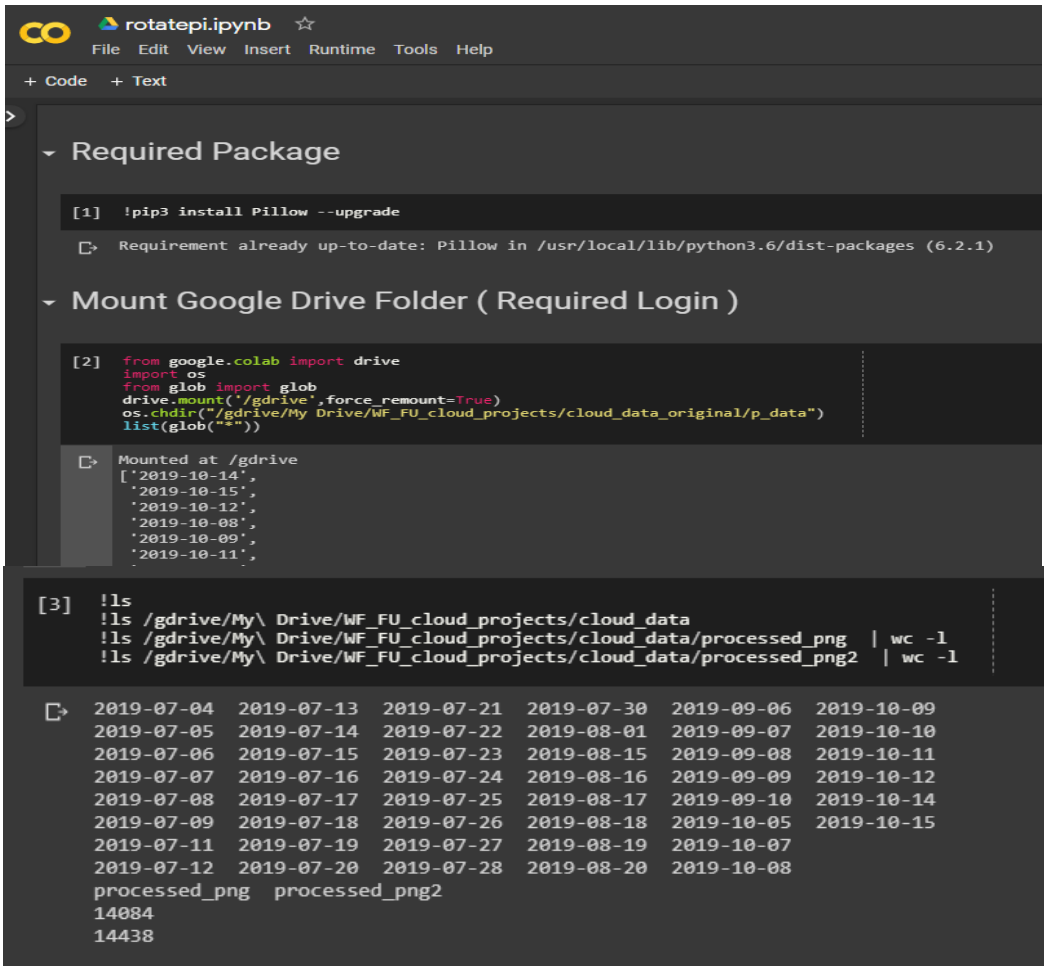
（1） 用 colab 開啟 rotatepi.ipynb ，進行雲層特徵圖片的旋轉與複製。

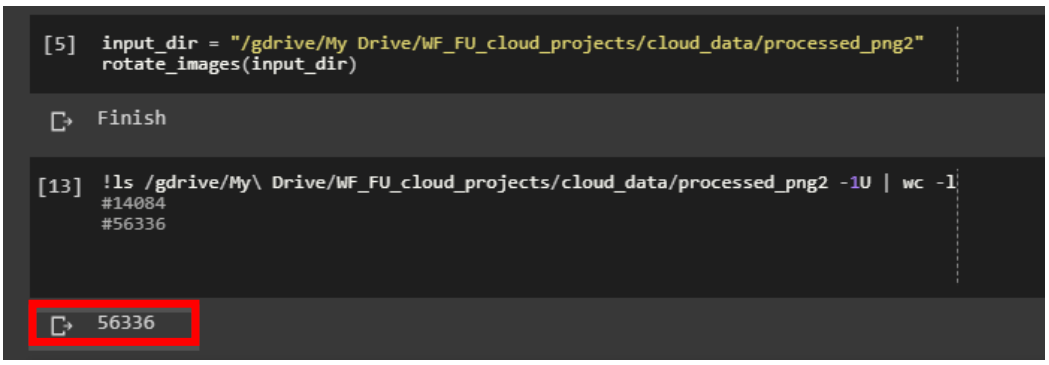
（2） 每張雲層特徵圖片會再產生三張分別轉 90、180 、270 度的雲層特徵圖片。

（3） 處理完成後會產生 4 倍的雲層圖片。

（4） 篩選夜晚或品質不好的相片。

（5） 程式執行如圖二十五。






圖二十五: 旋轉圖片增量程式執行程序示意

## 七、神經網路 CNN 模型的建立與訓練

（一）探討如何把處理好的資料輸入機器學習模組中，進行模型訓練，以預測接下來的 30 分鐘內降雨的機率。

1. 根據八二法則的原理，做模型訓練時會將資料以 8：2 的比例分爲 80%的訓練資料以及 20%測試資料如表六所示。
2. 表六所示的資料數是已經經過特徵提取流程，所以其資料數因旋轉後會是原本的 4 倍。
3. 此外，避免訓練和測試資料標籤的比例失調，在隨機打亂順序的時候會確保最多資料的沒有降雨的類別標籤比例維持在 50% 左右。

表六: 雲層圖片資料統計

| 資料類型 | 沒有降雨 | 可能降雨 | 降雨 | 資料總數 |
|---|---|---|---|---|
| 訓練資料 | 38157 | 4912 | 2000 | 45069 (80%) |
| 測試資料 | 9539 | 1228 | 500 | 11267 (20%) |
| 資料總數 | 47696 | 6140 | 2500 | 56336 (100%) |

(二)模型參數說明

1. 模型訓練中採用的是 CNN 模型，這個模型是參考 Pytorch 和 Keras 官方的 Image Captioning 的範例程式碼為基礎做修改，本研究的輸入資料是雲層圖片與大氣資料。
2. 整體流程可以參考圖二十六的流程系統，而模型的相關參數，整理如表七，該參數設定除了大部分都是參考原本 CNN 範例的參數，其中為配合 RGB 通道差值特徵向量的大小圖片尺寸修改成 256x256。

表七:訓練模型參數

| 參數 | 參數值 | 說明 |
|---|---|---|
| Crop_size | 256 像素 x256 像素 | 圖片大小 |
| Features_Size | 256FloatTensor | 圖片特徵向量大小 |
| Batch_size | 128 筆 | 批次大小(BatchSize) |
| Hidden_size | 16,64,256 個 | 隱藏層節點數量(HiddenSize) |
| num_layers | 1 層 | MLP 層數 |
| num_epochs | 5 次 | 迭代數(Epochs) |
| learning—rate | 0·001% | Learning Rate |
| activation | Relu | 激勵函數 |
| optimizer | Adam | 優化器 |

（三）實驗步驟：

1. 撰寫卷積神經網路訓練模型程式碼 CNN-Com.ipynb (用 CNN 建立模型與訓練)。
2. 用 colab 開啟產生 CNN model ，模型建立流程如圖二十六。
3. 產生多個 train model(附檔名為 ckpt)並測試 model 。


圖二十六: 模型建立流程

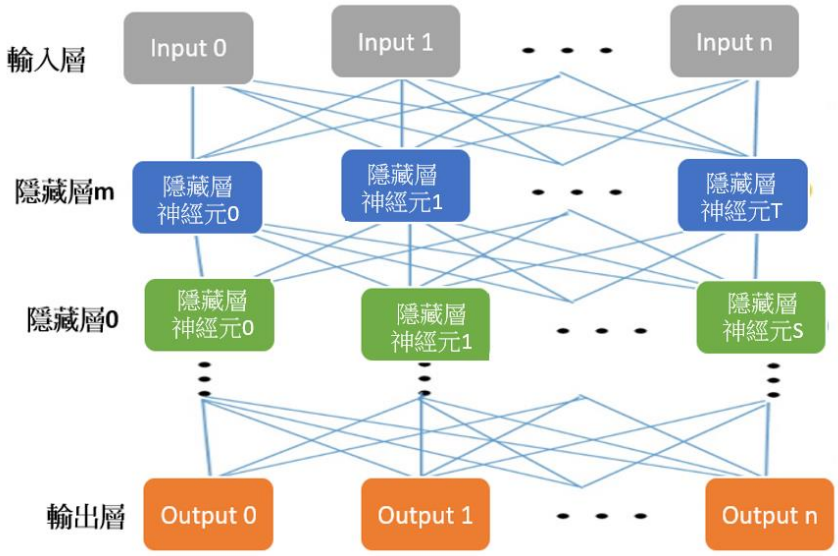
（四）實驗結果

1. 機器學習訓練需花費較多的時間，本研究不斷的測試與訓練，讓機器經過不斷的訓練並修正模型，得到初步的預測模型。
2. 圖二十七為隨機抓 10 筆雲層照片來預測未來 30 分鐘下雨的機率，類型包含不會下雨;可能會下雨和會下雨的預測機率。下雨預測機率為 $a\times10^{n}$ 的機率科學記號表示法，其中不會下雨的機率+可能會下雨的機率+會下雨的機率=1。


圖二十七: ＣＮＮ模型訓練與預測結果範例

## 八、隱藏層的層數對模型準確度的探討

（一） 因為隱藏層可以有多層架構所以稱為深度學習。而隱藏層的節點數或層數是 MLP 的關鍵參數，太少層數的模型可能成效不佳，太多層數的神經網路又可能有過擬合（over-fitting）的問題，這議題一直是機器學習過程中最困擾的地方，如圖二十八。




圖二十八: ＣＮＮ模型與隱藏層架構

（二） 模型訓練的過程，是利用樣本的評價作為類神經網路的反饋，調整各層神經元之間的權重參數，就像人類的大腦在學習一個新技能一樣，透過反覆嘗試錯誤，在神經層之間強化某些連結、弱化某些連結，最終把這項技能的邏輯，用一群神經層之間連結有強有弱的方式表達出來。

（三） 本研究在訓練模型過程中，希望進一步分析隱藏層層數對本模型預測成效是否會有影響，於是分別訓練隱藏層 16 層、64 層、256 層等的模型以比較其差異性。

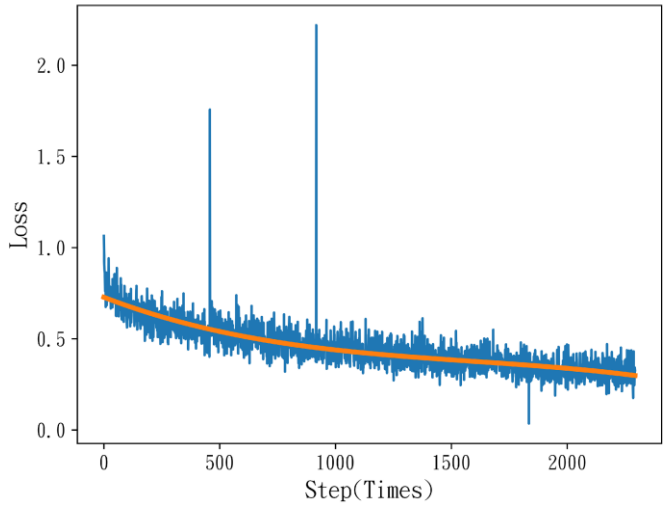
# 伍、研究結果

一、探討如何驗證模型的準確度

本研究的訓練模型最後需要透過測試來驗證準確度，而我們主要利用了幾種方法，分別是 Loss 曲線、精密度、召回率、F-measure 以及混淆矩陣進行分析。

（一）Loss 曲線之分析

1. 損失曲線是用來輔助機器學習模型中，參數是否設定合宜的判斷準則，可做為模型好壞的評估，並且會以「數值化」的方式告訴我們 model 的成效。損失(loss)是一個數值，表示對於單個樣本而言模型預測的準確程度。如果模型的預測完全正確，則損失為零，否則損失會較大。
2. 藍色的線條即特別多上下抖動的線條為原始的 loss 曲線，而中間橘色即平滑的線條為平滑化的 loss 曲線，以方便觀察其 loss 的變化情況。
3. 原始的 loss 顯示出一些較高或較低的 loss 的情況，但是不影響收斂的過程，對照平滑化的 loss 曲線，可以看出模型有逐漸收斂的情況，證明模型有在正常的學習如圖二十九。




圖二十九: 訓練過程 Loss 曲線

4. 我們也可以結合上圖與模型測試的準確度，來判斷是否過擬合(loss 曲線下降，準確度也下降)。如果 loss 趨勢是往下收斂且準確度有往上提升，表示這個模型是有成效的。

（二）模型成效之驗證

1. 各個參數會以要驗證的標籤為正樣本，其他標籤為負樣本的方式進行計算。從表八可以看到，除了可能會降雨的標籤驗證參數比較低以外，其他的標籤都有超過80%。

表八:模型驗證參數

| 驗證參數 | 不會降雨 | 可能會降雨 | 降雨 | 平均 |
|---|---|---|---|---|
| 精密度(Precision) | 85.217% | 46.211% | 83.317% | 71.582% |
| 召回率(Recall) | 86.012% | 53.218% | 80.981% | 73.070% |
| F-Measure | 85.160% | 49.811% | 81.512% | 72.161% |

2. 其中，各標籤驗證參數比率，是以未列入訓練的 20%原始資料當作測試資料，來進行預測後與降雨標記做比較，所整理出來的參數。測試資料相關的筆數描述在表六。
3. 從表八可以看出，不會降雨和會降雨標籤有著不錯的分類結果，而可能會降雨的標籤由於可能設定的臨界範圍過大，所以判別上會有些模糊地帶，相對的其訓練過程造成誤判，所以造成其分類相較其他來的低，但是這並不代表其標籤沒有鑑別能力。

**二、隱藏層的層數對模型準確度的分析**

（一）本研究在訓練模型過程中，想進一步分析隱藏層層數對本模型成效的影響，於是分別訓練了隱藏層 16 層、64 層、256 層等的模型。

（二）各隱藏層數之模型，分別分析其精密度、召回度與準確度，驗證模型之成效結果如表九。

（三）隱藏層數若太少，模型可能不夠精準，而具有太多層級的神經網絡可能過度擬合(overfitting)數據，導致對測試數據的解釋較差。根據下表，本研究可以了解 256 層的準確度高於 16 層和 64 層，準確度可達 80%以上。

（四）當然如果進一步提高隱藏層數也許可繼續提升準確度，但對於訓練的時間成本也會增加許多 。

表九: 各種不同隱藏層數 CNN 模型比較

| 模型類型 (hidden layer) | 平均精密度 Precision | 平均召回率 Recall | 平均 F-Measure | 準確度 Accuracy |
|---|---|---|---|---|
| CNN-16layer | 65.189% | 66.225% | 63.215% | 66.234% |
| CNN-64layer | 76.425% | 77.134% | 76.343% | 77.322% |
| CNN-256layer | 81.022% | 80.345% | 80.128% | 80.248% |

## 三、即時預測模型的應用與驗證

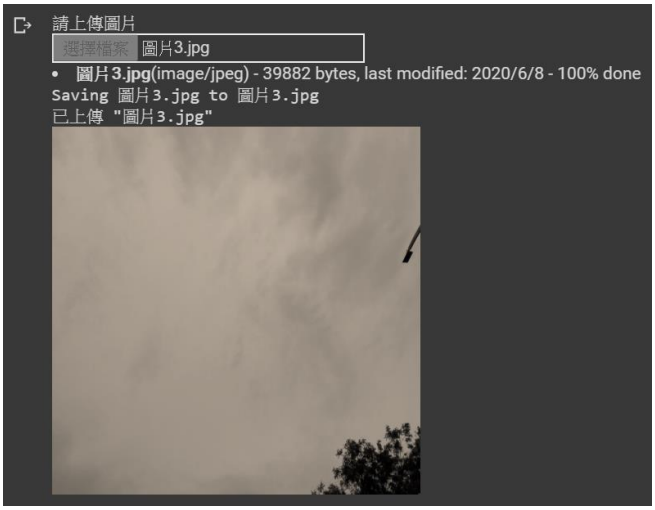
（一） 本研究由以上實驗訓練出一 CNN 神經網路模型，並進一步設計一隱藏層為 64 層的簡易預測模型，可即時針對雲層圖片來做未來 30 分鐘後降雨機率的預測。簡易模型可更快速做出預測但會犧牲一點準確度。

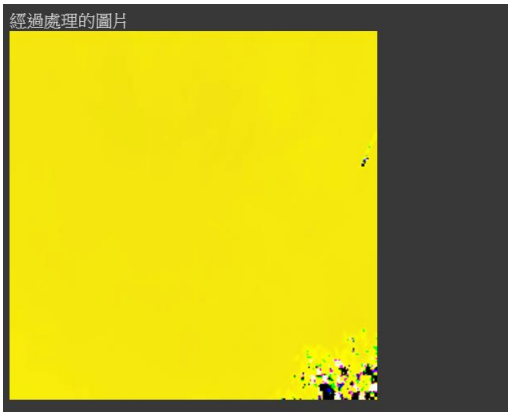
（二） 隨機給予本預測模型一張雲層圖片，本模型可在約 1 秒的時間內運算，即時預測未來 30 分鐘後可能的降雨狀況， 為配合一般人使用與判斷的實際習慣，本預測模型修改為只呈現出不會降雨與會降雨的預測機率與結果判斷，模型介面如圖三十。

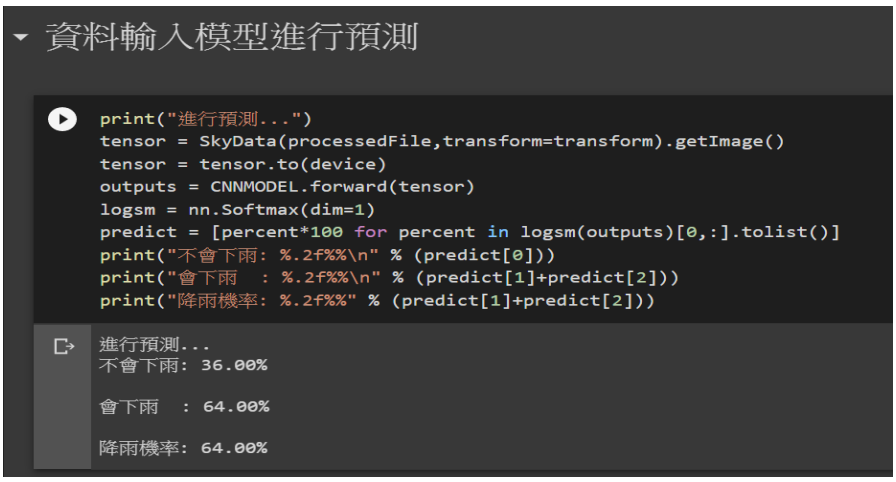
```
print("請上傳圖片")

uploaded = files.upload()
processedFile = None
orgFile = None

# only process one file
for filename in uploaded.keys():
  print('已上傳 "{}"'.format(filename))
  display(Image(filename,width=300, height=300))
  extension = Path(filename).suffix
  processedFile = os.path.join(tmppath,str(int(datetime.now().timestamp()))+".png")
  newImg = image_processing(filename, crop_size)
  newImg.save(processedFile, "PNG")
  print("\n經過處理的圖片")
  display(Image(processedFile,width=300, height=300))
  break
```






圖三十: 預測模型結果示意範例

（三） 每年五、六月是台灣的梅雨季節，配合這段時間降雨機率較高，本研究利用手機每日持續收集近一個月的雲層圖片且記錄 30 分鐘後降雨情況，並以本預測模型進行預測做比較驗證。

（四） 驗證結果整理如圖三十一之混淆矩陣，收集之雲層圖片共 167 張，其中 91 張 30 分鐘後沒有降雨，76 張有下雨。

| 預測類別 \ 真實類別 | 不會降雨 | 會降雨 |
|---|---|---|
| 不會降雨 | 83 | 29 |
| 會降雨 | 8 | 47 |

圖三十一: 預測模型之混淆矩陣

（五） 以圖三十一之混淆矩陣計算各類驗證參數如表十，簡易預測模型的整體準確度達到 77.84% ，各類驗證參數平均值也皆超過七成，和訓練模型之驗證結果相符。其中僅會降雨之召回率仍嫌不足，可能原因也許在訓練資料降雨雲層圖片資料筆數還有增加的空間，未來可繼續收集各類雲層圖片以進一步訓練修正模型。

**表十:預測模型驗證參數分析表**

| 類型 | 不會降雨 | 會降雨 | 平均值 |
|---|---|---|---|
| 準確度 | | 77.84% | |
| 精密度 | 74.11% | 85.45% | 79.78% |
| 召回率 | 91.21% | 61.84% | 76.53% |
| F-measure | 81.77% | 71.76% | 78.12% |

# 陸、結論

一、 通常一般天氣預測均需要相當高的成本與複雜的裝置，本研究提出了一個使用較低成本的 Arduino 面板、Arduino 感測器及樹莓派等物聯網裝置，來建置一個小型氣象觀測站的方法。這方法不但能有效蒐集溫濕度、大氣壓力、UV、降雨狀況與雲層圖片等參數，與一般天氣預測的方法相比，其他成本相對減少許多。由於整體裝置體積小，攜帶方便，不像傳統氣象站必須考慮很多架設條件，因此具有在不同地點能迅速架設的優勢。

二、 本研究經過數個月的時間，利用 Arduino 與樹莓派的相機模組，定時不斷地蒐集到足量的大氣資料與天空雲層圖片。透過本實驗的裝置及方法，並利用 Google 提供的 Colab 平台，對雲層資料做初步的處理、特徵提取與增量，即能有效訓練出特製的模型而加以應用，不需要額外複雜昂貴的處理環境，相較之下，也能更有效、更迅速地推廣給一般使用者使用。

三、 本研究的主要特色是先將這些大量的雲層圖片經過一連串的處理流程。把蒐集的圖片做一簡易的低解析度轉換，可以減少電腦的運算量。透過 RGB 通道差值的運算先初步提取雲層特徵圖，讓模型訓練更有效率。並以照片轉置的方法獲取更大的資料量供機器做學習。透過 CNN 卷積神經網路的機器學習方法，建立一套準確性相當高的辨識雲層圖片與降雨預測的模型。

四、 本研究展示了模型的各種驗證結果，包含準確度、精密度、召回率和 F-measure，證明本模型的可行性以及正確性，並且準確度達到八成以上，可以了解本研究提出的方法，對未來天氣的預測提供了另一個可能性。

五、 本研究同時進一步實驗，利用不同隱藏層層數的 CNN 模型進行訓練，隱藏層層數越多，越可提高模型的預測成效，但同時需衡量訓練時間成本增加與過擬合的問題。發現經訓練在 256 層時，其準確度即可達到 80%以上。因此本研究提出了一個有效訓練參數，並有效對未來天氣進行預測的方法。

六、 本研究訓練好的模型可以隨時進行預測，給予模型一張雲層圖片，即可在短時間內得到針對此小區域範圍的降雨類型預測結果，相較氣象局之衛星預測多針對大範圍預估來得準確，也即時許多。

七、 總結以上探討，本研究提出了一套方法，利用地面往天空拍攝的雲層圖像，及氣象資料進行機器學習訓練，藉由訓練出來的模型可針對小區域範圍未來 30 分鐘後降雨的可能型態做預測。採用本研究同樣的模型建構方法，可套用各種不同地型蒐集的資料

做訓練，例如都市與鄉村、迎風面與背風面、山上或海邊等的降雨預測模型，都可客製化做訓練以符合需求。

# 柒、參考資料


一、交通部中央氣象局一周天氣預測。

網址：https://www.cwb.gov.tw/V8/C/W/W50_index.html

二、氣象站維基百科。

網址：https://zh.wikipedia.org/zh-tw/氣象站

三、雲的故事

網址：http://ast.nhps.tp.edu.tw/home/sally/Source/Unit4-2/Weather/cloud.htm

四、一文看懂四種基本的神經網絡架構。

網址：https://kknews.cc/zh-tw/code/2z4lmoe.html

五、常見監督式機器學習演算法 – 機器學習兩大學習方法。

網址：https://blog.gcp.expert/supervised-learning-classification-regression-algorithms/

六、強化學習系列（五）：蒙特卡羅方法（Monte Carlo）。

網址：https://www.itread01.com/content/1548393874.html

七、神經網路和深度學習入門知識。

網址：https://www.itread01.com/content/1546909226.html

八、神經網路(多層感知機 Multilayer perceptron, MLP)

網址：https://www.itread01.com/content/1542288247.html

九、卷積神經網路的運作原理。 網址：

https://brohrer.mcknote.com/zhHant/how_machine_learning_works/how_convolutional_neural_networks_work.html

十、機器學習\統計方法: 模型評估-驗證指標(validation index)

網址：https://medium.com/@chih.sheng.huang821/機器學習-統計方法-模型評估-驗證指標-b03825ff0814

十一、A. Kazantzidis, P. Tzoumanikas, A. Bais, S. Fotopoulos, and G. Economou, “Cloud detection and classification with the use of whole-sky ground-based images,” Atmospheric Research, vol. 113, pp. 80 – 88, 2012.

十二、使用機器學習預測天氣:

網址：https://codertw.com/人工智慧/4568/ (如何使用 TensorFlow 高階 API Estimator 子類

DNNRegressor，建立了一個模型，根據收集的數字特徵預測第二天的平均溫度。)

十三、深度學習：CNN 原理

網址：https://medium.com/@CinnamonAITaiwan/深度學習-cnn 原理-keras 實現-432fd9ea4935

十四、基於機器學習的數值天氣預報風速訂正計畫:

網址：http://qxqk.nmc.cn/html/2019/3/20190312.html (利用機器學習進行風速之預測。)

十五、利用人工智慧來有效且準確的預測天氣:

網址：https://www.itritech.net/blog/climate-forecast-seek-ai-for-help/ (氣象衛星撈巨量資料，天氣預測 AI 來幫忙。)

十六、機器學習的數值天氣預報風速訂正計畫:

網址：https://www.inside.com.tw/article/6748-what-is-machine-learning (機器學習技術概念。)

## 【評語】032806

作品使用感測器搜集大氣資訊，並以攝影機拍攝雲層相片，再用人工智慧方法結合這些資訊預測卅分鐘後該地區是否會下雨，可以達到相當準確度，是機器學習與人工智慧技術的有趣應用。作品搜集大量資料，並能使用最新技術，相當值得鼓勵。訓練的影像特徵與降雨的直接關聯度可能會受季節、雨水感測器的殘留誤差等影響，可以在預測準確度誤差分析方面再多著墨，另外亦可考慮加長預測時間，以提高實用性。

## 作品海報

## 壹、研究動機

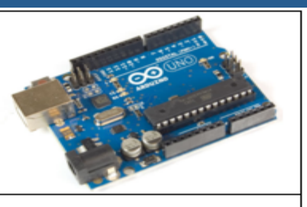
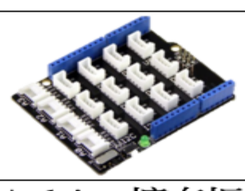
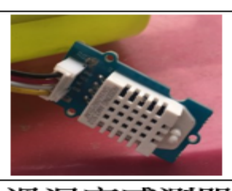
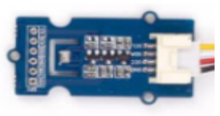
人類的生活受到不同的天氣變化而影響。現有的天氣預測僅提供大區域範圍的普遍預測資料，為能針對小範圍區域提供即時和確切的天氣預報，本研究提出一套以大數據、AI人工智慧及配合物聯網蒐集資料的機器訓練模型方法，來建立一個可預測未來30分鐘小區域範圍的降雨預測模型，最後基於機器學習模型來預測天氣，並進一步驗證本天氣預測模型的準確度。

## 貳、研究目的

一、探討如何利用Arduino 的各種感測器，蒐集溫度、 濕度、大氣壓力、UV、以及是否降雨等資料。
二、探討如何使用樹莓派(Raspberry Pi) 相機模組拍攝蒐集雲層圖片，作為模型預測之主要輸入資料。
三、探討如何將大量的雲層圖片進行特徵值提取，以建立並訓練模型。
四、探討如何驗證訓練模型的準確度。
五、探討利用卷積神經網路不同隱藏層數量，對模型預測降雨準確度的影響。
六、建構並實際運用一套可即時預測降雨狀況的預測模型。

## 參、研究設備與器材

一、硬體設備:

| | | |
|---|---|---|
|  |  |  |
| 樹莓派 pi 3B | 樹莓派相機模組 | Arduino UNO |
|  |  |  |
| Arduino 擴充板 | 溫濕度感測器 | 大氣氣壓感測器 |
|  |  |  |
| UV 感測器 | 雨滴感測器 | 資料蒐集伺服器 |

二、軟體設備:

| 方法 | 環境/工具 | 說明 |
|---|---|---|
| 機器學習框架 | Keras | Python3 |
| 程式開發環境 | Google Colaboratory | GPU tesla T4 |
| OS | Google 線上環境 | Ubuntu 18.04 |

## 肆、研究過程與方法

### 一、卷積神經網路(CNN)

卷積神經網路(CNN)是由多個卷積層(Convolution Layer)，池化層(Pooling Layer)，全連結層(Fully Connected Layer)、線性整流元(Rectified Linear Unit, ReLU) 等元件組成。在圖片辨識和聲音辨識上都有著非常出色的表現。本研究利用其圖片辨識能力，應用於雲層圖片辨識。




### 二、分類模型驗證

1.混淆矩陣(Confusion Matrix): 混淆矩陣分爲真實類別(True Class) 也就是正確的標籤、以及預測類別(Predicted Class) 即預測的標籤。這兩類標籤分別都有正樣本(Positive) 以及負樣本(Negative)。

| | | 真實類別(True class) | |
|---|---|---|---|
| | | 正樣本(Positive) | 負樣本(Negative) |
| 預測類別 (Predicted class) | 正樣本(Positive) | True Positive(TP) | False Positive(FP) |
| | 負樣本(Negative) | False Negative(FN) | True Negative(TN) |

2.精密度(Precision)：精密度是指在預測類別中為正樣本，而真實類別也是正樣本的正確的比例。公式(1) 為精密度的計算方式。

$$\text{精密度}=\frac{TP}{TP+FP} \quad \text{公式(1)}$$

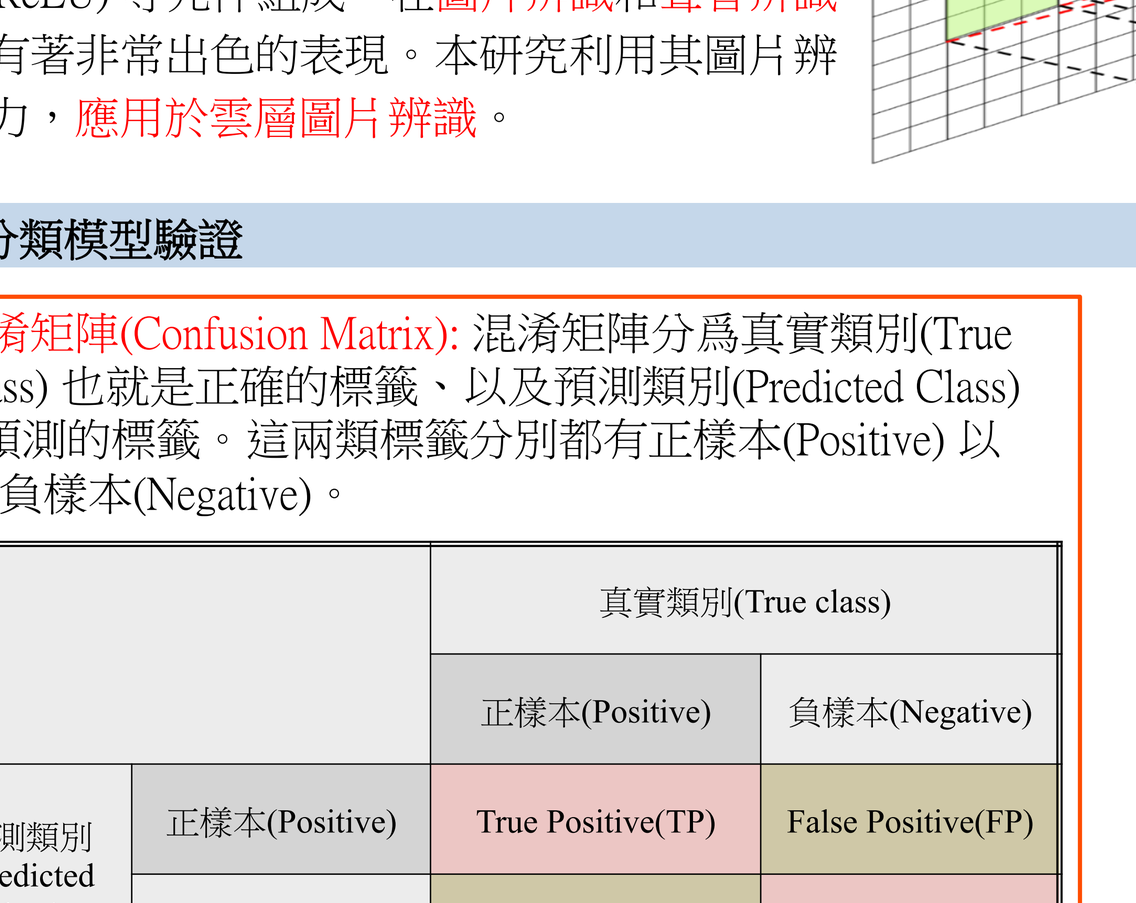
3.召回率(Recall)：召回率是計算真實類別之正樣本被成功預測出來的機率，其公式如公式(2) 所示，計算結果越高越好。

$$\text{召回率}=\frac{TP}{TP+FN} \quad \text{公式(2)}$$

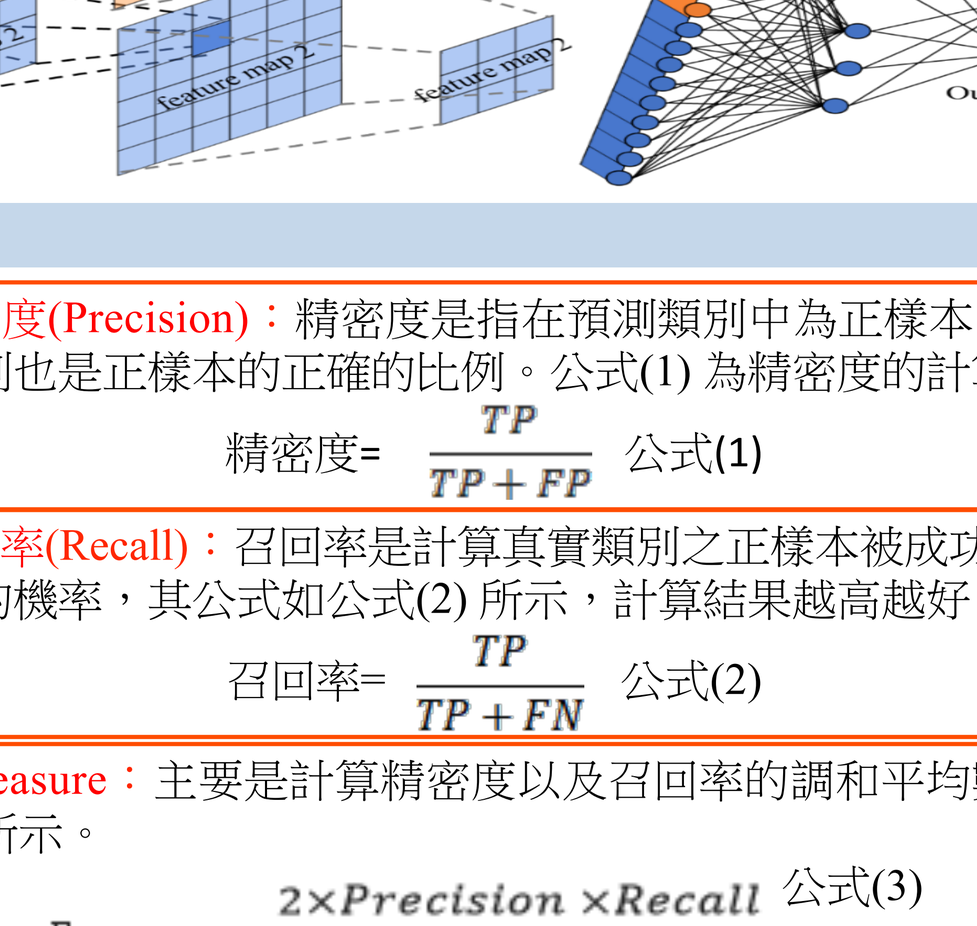
4.F-measure：主要是計算精密度以及召回率的調和平均數公式如(3)所示。

$$\text{F-measure}=\frac{2\times Precision\times Recall}{Precision+Recall} \quad \text{公式(3)}$$

### 三、研究架構概述：小型氣象站系統與機器學習架構



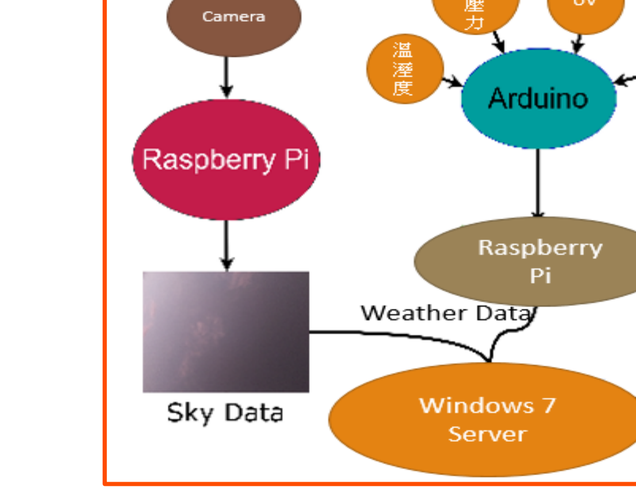

本實驗相關儀器裝置規劃圖



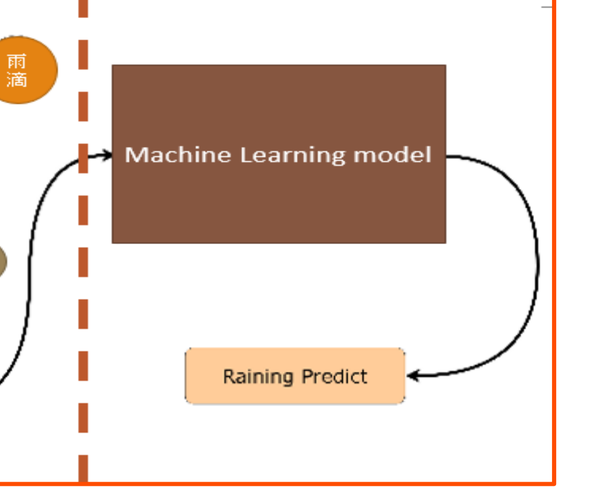

本實驗氣象資料蒐集站裝置圖

### 四、探討如何利用Arduino 與各式感測器，建立小型氣象資料蒐集站

（一）說明:本研究利用Arduino結合各種感測器模組，來建構氣象資料蒐集站，搜集大氣資料。




不同感測器偵測數據情形



相機模組拍攝雲層畫面

（二）本研究採用的電子元件整理如下表。

| 設備名稱 | 設備型號 | 功能 |
|---|---|---|
| 樹莓派 Pi 3 | 3B | 將圖像與感測器資料傳回伺服器 |
| 相機模組 | Noir module v2 | 拍攝雲層圖片 |
| Arduino | UNO | 將感測器資料傳回給樹莓派 |
| Arduino 擴充版 | Base shield Grove | 擴充感測器插槽 |
| 溫溼度感測器 | Grove Temperature Humidity sensor pro | 測量溫度與濕度 |
| 大氣壓力感測器 | Grove Barometer | 測量大氣壓力 |
| UV 感測器 | Grove UV sensor | 測量 UV 強度 |
| 雨滴感測器 | 水滴型雨水感測器 | 測量是否下雨 |

（三）實驗結果

| 雨滴感測數值 | 下雨狀態分類 |
|---|---|
| 0 ~ 341 | Not Raining |
| 342 ~ 682 | Rain Warning |
| 683 ~ 1023 | Raining |

雨滴感測器蒐集下雨狀況的數據

（四）實驗討論

1透過小型氣象站，我們蒐集了相當多筆的大氣資料，並利用網頁即時呈現相關數據。

2.由大氣相關儀器偵測的資料，並無法十分有效預測降雨狀況，所以本研究希望透過「機器學習」能建構有效的預測模型。

3.本研究依雨滴感測數值，建立降雨的參數為(1) 0~341為“不降雨”；(2) 342~682為“ 可能降雨” ；(3) 683~1023為“降雨” 。

蒐集的天氣資料：以10月13日為例

| 時間戳記 | 大氣壓力 (hPa) | 溫度 (˚C) | 濕度 (%) | 紫外線 ($mW/m^2$) | 下雨狀態 (Raining) |
|---|---|---|---|---|---|
| 2019/10/13 | 1006 | 26.2 | 94.4 | 0 | Rain Warning |
| 2019/10/13 | 1006 | 26.2 | 94.425 | 0 | Rain Warning |
| 2019/10/13 | 1006 | 26.2 | 94.5 | 0 | Raining |
| 2019/10/13 | 1006 | 26.2 | 94.5 | 0 | Rain Warning |
| 2019/10/13 | 1006 | 26.2 | 94.475 | 0 | Rain Warning |
| 2019/10/13 | 1005 | 26.5 | 93.6 | 18 | Not Raining |
| 2019/10/13 | 1005 | 26.5 | 93.48 | 22.4 | Not Raining |
| 2019/10/13 | 1005 | 26.5 | 93.35 | 37.25 | Not Raining |
| 2019/10/13 | 1005 | 26.52 | 93.3 | 51 | Not Raining |

## 五、探討如何使用樹莓派(Raspberry Pi) 相機模組作為雲層拍攝的裝置，蒐集雲層圖片作為預測資料

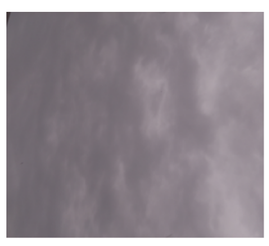
（一）雲層拍攝裝置

（三）蒐集的雲層圖片與氣象資料統計表

| 資料類型 | 有效資料/無效資料 | 說明 |
|---|---|---|
| 雲層圖片 | 14084/11029 | - |
| 氣象感測器資料 | 16314/105867 | 沒有降雨: 11989<br>接近降雨: 2789<br>降雨:1536 |
| 可用訓練資料 | 14084 | 沒有降雨: 11924<br>接近降雨:1535<br>降雨:625 |

（二）實驗結果

蒐集的雲層與氣象資料整理表。

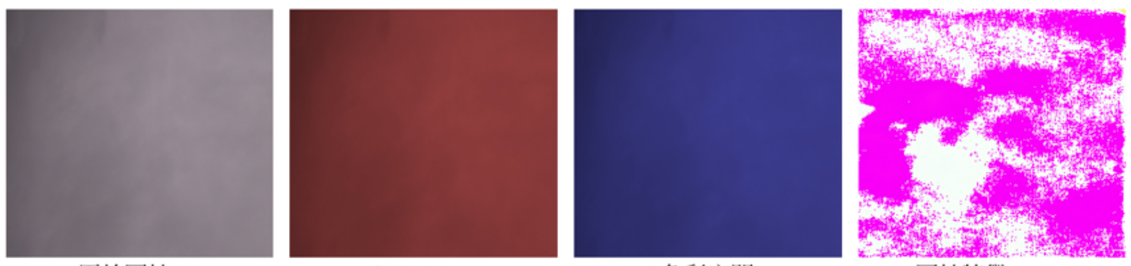
| | | |
|---|---|---|
| 時間 | 2019/10/11 12:00 |  |
| 大氣壓力 | 1016 | |
| 溫度 | 29.65 | |
| 濕度 | 59.1 | |
| UV | 230.098 | |
| 下雨狀況 | Not Raining | |
| 時間戳記 | 2019/08/19 10:00 |  |
| 大氣壓力 | 1020 | |
| 溫度 | 26.51 | |
| 濕度 | 88.714 | |
| UV | 56.324 | |
| 下雨狀況 | Rain Warning | |
| 時間 | 2019/08/16 10:00 |  |
| 大氣壓力 | 1017 | |
| 溫度 | 27.248 | |
| 濕度 | 85.294 | |
| UV | 15.242 | |
| 下雨狀況 | Raining | |

## 六、雲層圖片的特徵提取與合併大氣資料

（一）說明

當雲層發生變化時，環境之原有色彩會遭到雲層的掩蓋而呈現灰階色彩。而灰階色彩具有一項重要的RGB 特徵，那就是三原色的強度相等 。當RGB 之差值越小，其組成越接近灰階色彩，也就代表越有可能存在雲層。

本研究在機器學習前，將先做雲層特徵的提取，以利機器學習，而提取雲層特徵較佳的方式，即是先進行RGB通道差值的運算，轉換原始圖片為RGB通道差值來保存，以做為CNN卷積神經網路的原始輸入資料 。



原始圖片　R色彩空間　B色彩空間　圖片特徵

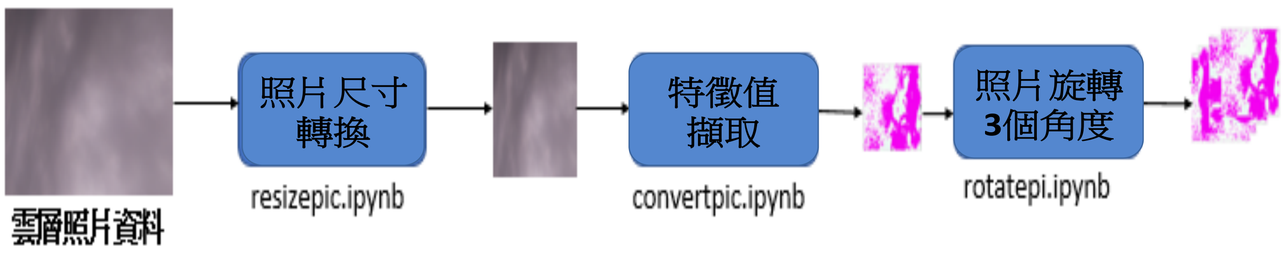
（三）實驗結果

處理資料數量為56336張(含3次旋轉的照片數)，以8：2 的比例，分成訓練資料及測試資料，如下表所示。

| 資料類型 | 沒有降雨 | 可能降雨 | 降雨 | 資料總數 |
|---|---|---|---|---|
| 訓練資料 | 38157 | 4912 | 2000 | 45069 (80%) |
| 測試資料 | 9539 | 1228 | 500 | 11267 (20%) |
| 資料總數 | 47696 | 6140 | 2500 | 56336 (100%) |

（二）實驗步驟





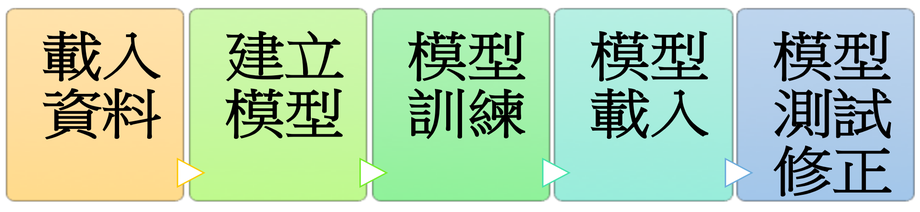

## 七、神經網路CNN模型的建立與訓練

（一）模型參數說明

本研究之模型訓練採用的是CNN模型，參考Pytorch 和Keras官方的Image Captioning 的範例程式碼為基礎做修改，輸入資料是雲層圖片與大氣資料。模型的相關參數，整理如下表，該參數設定除了大部分都是參考原本CNN範例的參數，其中為配合RGB通道差值特徵向量的大小圖片尺寸修改成256x256。

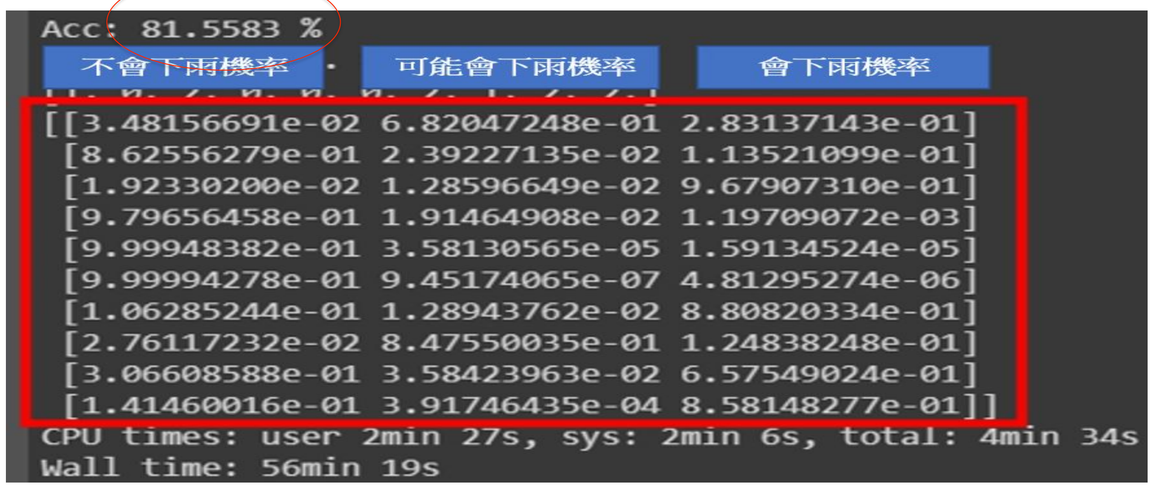
| 參數 | 參數值 | 說明 |
|---|---|---|
| Crop_size | 256 像素 x256 像素 | 圖片大小 |
| Features_Size | 256FloatTensor | 圖片特徵向量大小 |
| Batch_size | 128 筆 | 批次大小(BatchSize) |
| Hidden_size | 16,64,256 個 | 隱藏層節點數量(HiddenSize) |
| num_layers | 1 層 | MLP 層數 |
| num_epochs | 5 次 | 迭代數(Epochs) |
| learning—rate | 0·001% | Learning Rate |
| activation | Relu | 激勵函數 |
| optimizer | Adam | 優化器 |

（二）實驗步驟-模型建立流程




（三）實驗結果

下圖為隨機抓10筆雲層照片來預測未來30分鐘下雨的機率，類型包含不會下雨、可能會下雨和會下雨的預測機率。下雨預測機率為 $a\times10^n$ 的機率科學記號表示法，其中不會下雨的機率+可能會下雨的機率+會下雨的機率=1。

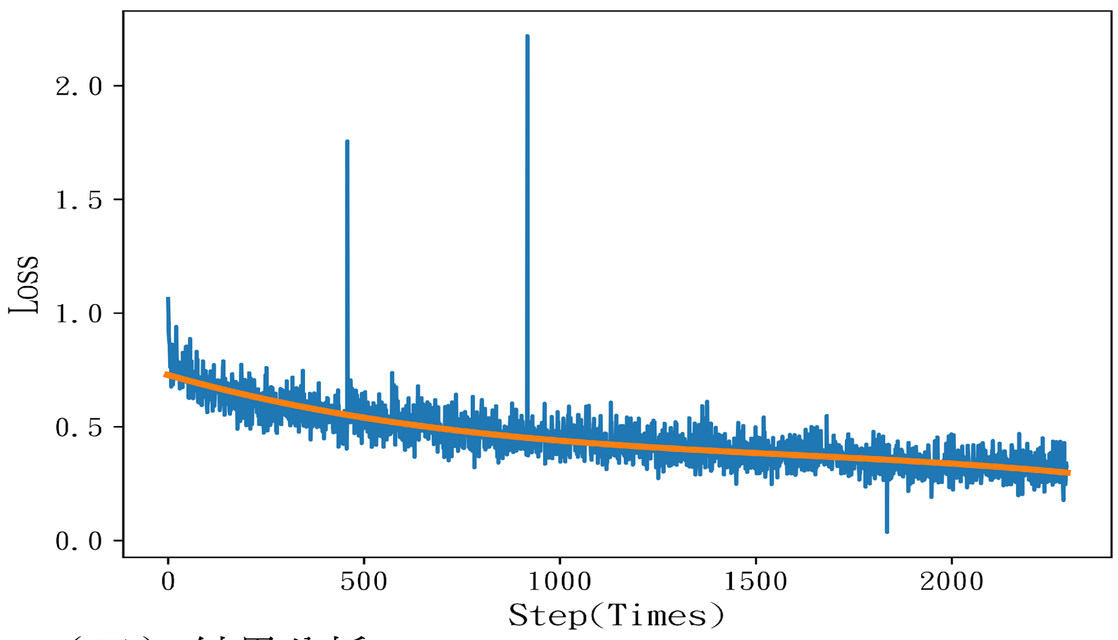


# 伍、研究結果

## 一、驗證模型的準確度

### （一） Loss曲線分析

中間橘色的線條為平滑化的loss 曲線，由此曲線可以看出模型有逐漸收斂的情況，證明模型有正常的學習。




### （二）模型驗證參數

ＣＮＮ模型以測試資料來進行參數分析，各個參數以要驗證的標籤為正樣本，其他標籤為負樣本的方式進行計算，得到以下結果。

| 驗證參數 | 不會降雨 | 可能會降雨 | 降雨 | 平均 |
|---|---|---|---|---|
| 精密度(Precision) | 85.217% | 46.211% | 83.317% | 71.582% |
| 召回率(Recall) | 86.012% | 53.218% | 80.981% | 73.070% |
| F-Measure | 85.160% | 49.811% | 81.512% | 72.161% |

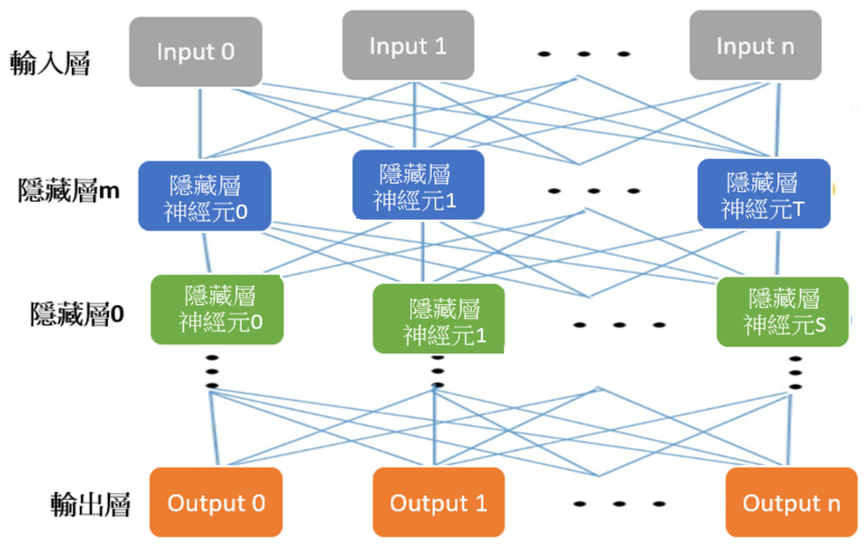
### （三） 結果分析

1.本實驗得到的loss曲線是往下收斂，且模型準確度有向上提升，表示這個模型訓練是有成效的。

2.從各驗證參數表中可以看到，除了可能會降雨的標籤驗證參數比較低以外，其他的標籤都有超過80%。

## 二、隱藏層的層數對模型準確度的分析

### （一）ＣＮＮ模型與隱藏層架構




### （二）各種不同隱藏層數CNN模型比較

| 模型類型 (hidden layer) | 平均精密度 Precision | 平均召回率 RecaII | 平均 F-Measure | 準確度 Accuracy |
|---|---|---|---|---|
| CNN-16layer | 65.189% | 66.225% | 63.215% | 66.234% |
| CNN-64layer | 76.425% | 77.134% | 76.343% | 77.322% |
| CNN-256layer | 81.022% | 80.345% | 80.128% | 80.248% |

### (三) 結果分析

1.隱藏層數若太少，模型較不精準，透過提高隱藏層層數可以提升準確度與各驗證參數。

2.本研究發現在256層的準確度除高於16層和64層外，準確度更可達80%以上。

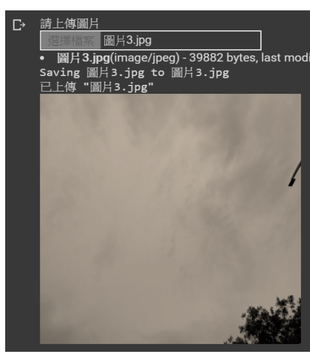
## 三、即時預測模型與應用

```
print("請上傳圖片")

uploaded = files.upload()
processedFile = None
orgFile = None

# only process one file
for filename in uploaded.keys():
  print('已上傳 "{}"'.format(filename))
  display(Image(filename,width=300, height=300))
  extension = Path(filename).suffix
  processedFile = os.path.join(tmppath,str(int(datetime.now().timestamp()))+".png")
  newImg = image_processing(filename, crop_size)
  newImg.save(processedFile, "PNG")
  print("\n經過處理的圖片")
  display(Image(processedFile,width=300, height=300))
  break
```

```
請上傳圖片
選擇檔案 圖片3.jpg
• 圖片3.jpg(image/jpeg) - 39882 bytes, last modified: 2020/6/8 - 100% done
Saving 圖片3.jpg to 圖片3.jpg
已上傳 "圖片3.jpg"
```



```
資料輸入模型進行預測

print("進行預測...")
tensor = SkyData(processedFile,transform=transform).getImage()
tensor = tensor.to(device)
outputs = CNNMODEL.forward(tensor)
logsm = nn.Softmax(dim=1)
predict = [percent*100 for percent in logsm(outputs)[0,:].tolist()]
print("不會下雨: %.2f%%\n" % (predict[0]))
print("會下雨  : %.2f%%\n" % (predict[1]+predict[2]))
print("降雨機率: %.2f%%" % (predict[1]+predict[2]))

進行預測...
不會下雨: 36.00%

會下雨  : 64.00%

降雨機率: 64.00%
```

真實類別

| 預測類別 | 不會降雨 | 會降雨 |
|---|---|---|
| 不會降雨 | 234 | 56 |
| 會降雨 | 47 | 187 |

| 類型 | 不會降雨 | 會降雨 | 平均 |
|---|---|---|---|
| 準確度 | | 80.34% | |
| 精密度 | 80.69% | 79.91% | 80.30% |
| 召回率 | 83.27% | 76.95% | 80.11% |
| F-measure | 81.96% | 78.41% | 80.21% |

（一）本研究以訓練之CNN神經網路模型，進一步設計一可即時針對雲層圖片來做未來降雨可能性的分類預測模型，即時預測模型運作範例如上。

（二）隨機給予一張雲層圖片，本模型可在約1秒的時間內運算，即時預測未來30分鐘後可能的降雨狀況。

（三）以台灣今年五、六月的梅雨季節來收集雲層圖片共524張，實際驗證後所得的混淆矩陣如上，預測的準確度為80.34%。

# 陸、結論

一、本研究提出了一個使用較低成本的Arduino 開發板、Arduino感測器及樹莓派相機等物聯網裝置，來建置一個小型氣象觀測站的方法。這方法不但能有效蒐集溫濕度、大氣壓力、ＵＶ、降雨狀況與雲層圖片等資料，由於整體裝置體積小，攜帶方便，具有在不同地點能迅速架設的優勢。

二、透過本實驗的裝置及方法，並利用Google提供的Colab平台，對雲層資料做尺寸轉換、特徵提取與增量，即能有效訓練出特製的模型而加以應用，不需要額外複雜昂貴的處理環境，相較之下，能更有效、更迅速地推廣給一般使用者使用。

三、把蒐集的圖片做一簡易的低解析度轉換，可以減少電腦的運算量。透過RGB通道差值的運算先初步提取雲層特徵圖，可讓模型訓練更有效率。並以照片轉置的方法獲取更大的資料量供機器做學習。透過CNN卷積神經網路的機器學習方法，可建立一套準確性相當高的辨識雲層圖片與降雨預測的模型。

四、本研究展示了模型的各種驗證結果，包含準確度、精密度、召回率和F-measure，證明本模型的可行性以及正確性，並且準確度達到80%以上。

五、本研究同時進一步實驗，針對不同隱藏層層數的CNN模型進行訓練，發現隱藏層層數越多，越可提高模型的預測成效，但同時需衡量訓練時間成本增加與過擬合的問題。本研究在256層隱藏層下，其準確度即可達到80%以上。

六、本研究訓練並設計出一即時預測模型可以隨時進行預測，即給予模型一張雲層圖片，可在短時間內得到針對此小區域範圍的降雨類型預測結果，相較氣象局之衛星預測多針對大範圍預估來得準確，也即時許多。

七、總結以上探討，本研究提出了一套方法，利用地面往天空拍攝的雲層圖像，及氣象資料進行機器學習訓練，藉由訓練出來的模型可針對小區域範圍未來30分鐘後降雨的可能型態做預測。採用本研究同樣的模型建構方法，可套用在各種不同地形蒐集的資料來做訓練，例如都市與鄉村、迎風面與背風面、山上或海邊等的降雨預測模型，都可客製化做訓練以符合需求。

# 柒、參考資料


一、雲的故事。網址：http://ast.nhps.tp.edu.tw/home/sally/Source/Unit4-2/Weather/cloud.htm

二、一文看懂四種基本的神經網絡架構。網址：https://kknews.cc/zh-tw/code/2z4lmoe.html

三、神經網路和深度學習入門知識。網址：https://www.itread01.com/content/1546909226.html

四、使用機器學習預測天氣。網址：https://codertw.com/人工智慧/4568/

五、深度學習：CNN原理。網址：https://medium.com/@CinnamonAITaiwan/深度學習-cnn原理-keras實現-432fd9ea4935

六、TensorFlow + Keras 深度學習人工智慧實務應用。林大貴著。

七、關於人工智慧一定要懂的96件事--圖解AI人工智慧大未來。三津村直貴著。陳子安譯。