

AI人臉辨識

組員：10640017A 張峻崕 10640041A 蔡俊昇
指導老師：葉建寧

摘要

隨著電腦科技發展的進步，加上為了提供給使用者更大的便利和可靠性，所以漸漸開始使用人臉辨識、語音辨識、指紋辨識、掌紋辨識和虹膜辨識等相關的生物認證技術。其中以人臉辨識最廣為被使用和研究，因為使用者並不需要穿戴額外的裝置，也不需要和受測裝置有任何接觸，只需要透過基本的攝影照相裝置，都可以得到辨識所需要的資料，其未來的運用範圍可以是養老院、遊樂場、大門進出口等等，可以運用到地方之多也非常的便利，而目前人臉辨識系統的方法有相當多種，每一種都有一定的準確率。本論文利用Haar-like特徵裡的Adaboost分類器來進行人臉辨識來判斷是否為人臉。

結果與討論

選取圖片，然後提取其特徵，運用 Haar-like 判斷是否為人臉，再運用 Adaboost 分類器進行分類

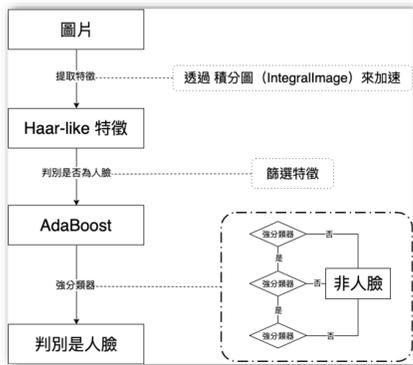


圖1.人臉辨識流程圖

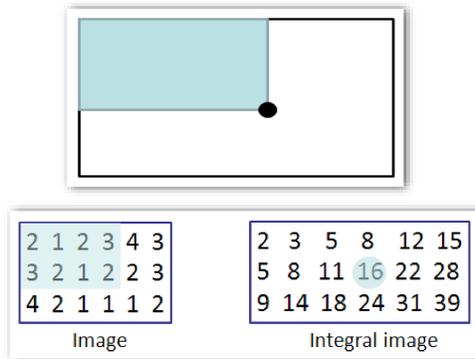
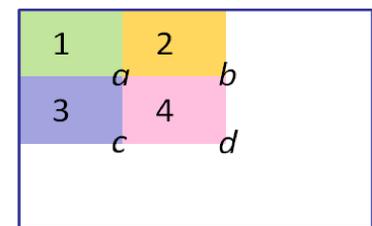


圖2.積分圖



$$\begin{aligned} a &= \text{sum}(1) & \text{Sum}(1) &= a \\ b &= \text{sum}(1+2) & \text{Sum}(2) &= b - a \\ c &= \text{sum}(1+3) & \text{Sum}(3) &= c - a \\ d &= \text{sum}(1+2+3+4) & \text{Sum}(4) &= d + a - b - c \end{aligned}$$

圖3.積分圖公式

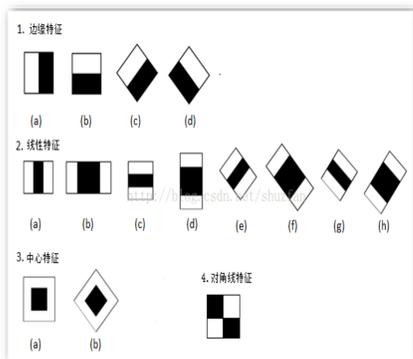


圖4.Haar特徵

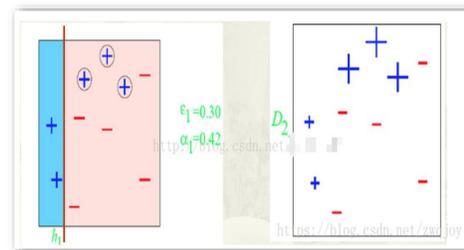


圖5.Adaboost步驟一

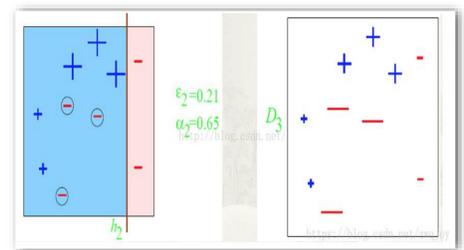


圖6.Adaboost步驟二

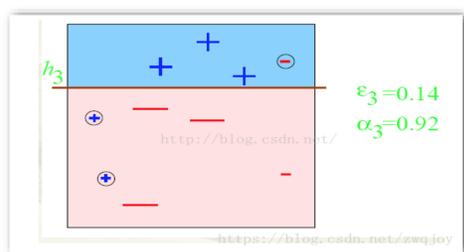


圖7.Adaboost步驟三

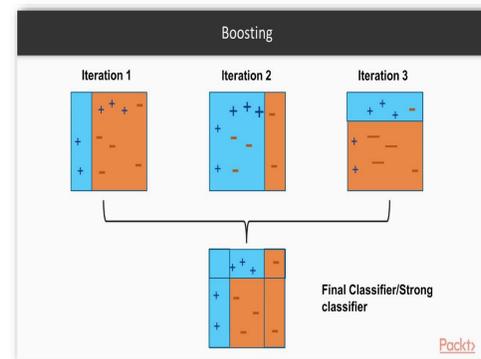


圖8.Adaboost方法

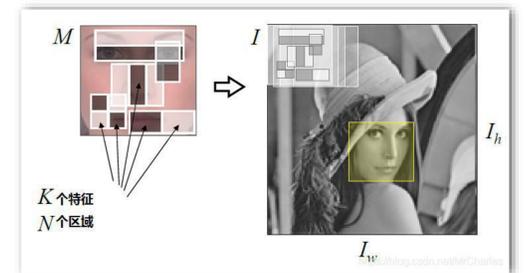


圖9.特徵提取

結論

在製作這個報告時遇到很多難題，雖然沒有做到很完善，還有許多地方需要加強，最主要的還是最後人臉辨識的框，還沒找到更小的程式碼數據，所以有時照片拍出來的，會有一些不是對照在臉上，但也因為這個讓我更了解人臉辨識的功能，如果能越做越好這樣子安全性一定會越好的。

參考資料

- <https://www.netadmin.com.tw/netadmin/zh-tw/technology/CDAFF5E13BEF4B0D8FF1199F43022D75>
- <https://www.itread01.com/content/1536736827.html>
- <http://alex-phd.blogspot.com/2014/03/haarhaar-adaboost.html>

下一個巴菲特

組員：吳孟恩 林亭秀 陳挺任 義守大學 財務與計算數學系
 指導老師：葉建寧 老師

簡介

近幾個月以來，因為新型冠狀病毒COVID-19的肆虐，導致全球股市波動的幅度增加。由於美國的疫情無法控制加上美國大選的因素，以至於美國股價起伏不定，所以我們想要設計一個能夠預測美國股價波動的程式。這份作業是利用網路上所得的資料來設計的，並非如此專業，所以以此項作業預測無法達到百分之百。但我們會盡所能完成此項作業，成為下一個巴菲特。

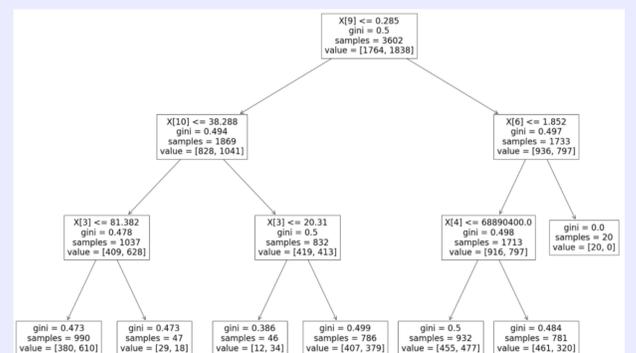
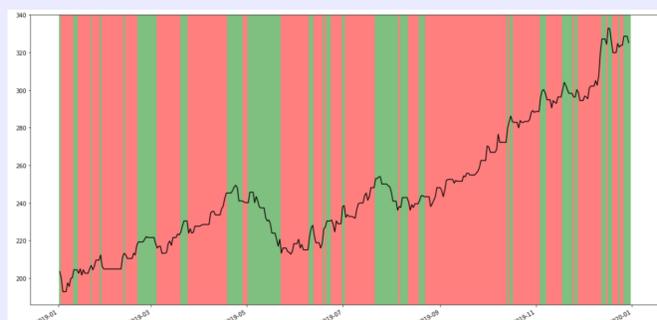
研究目的與目標

1. 設定預測目標。(做出來的AI要幫我解決什麼問題?)
2. 搜集資料數據。(大數據的時代，此為最關鍵的第一步。)
3. 選擇AI模型。(AI模型各有優缺，選出最適合的模型。)
4. 標記你的預測目標。(先準備一份標準答案，讓AI能分辨對錯。)
5. 整理剩餘的資料。(答案以外的數據，就是AI的學習資料庫。)
6. 把資料切成三份。(一份給AI學習，一份給AI改正，一份給AI測驗。)
7. 讓AI學習。(設定基礎參數，開始讓AI進行學習，然後生出一個模型。)
8. 讓AI改正。(驗證一下生出來的模型好不好，進行模型的調整。)
9. 讓AI測驗。(實際給AI，一批它沒有看過的資料，看看結果如何。)
10. 測驗結果分析。(對結果進行績效分析，此結果決定這個模型該保留與否。)
11. 模型保留->未來實際應用/模型不保留->重新建設。

實驗結果與展示

Date	Open	High	Low	Close	Volume
2000-01-04	36.650322	36.650322	36.032649	36.650322	200662321971
2000-01-05	36.650321	37.473900	36.238453	37.473900	402466776297
2000-01-06	37.268044	37.473901	36.650322	36.650322	197545701266
2000-01-07	35.414920	36.032641	35.003052	35.620773	235270327441
2000-01-10	36.650321	37.062028	36.032647	36.856174	276171665217
...
2021-01-04	530.000000	540.000000	528.000000	536.000000	38770328
2021-01-05	536.000000	542.000000	535.000000	542.000000	34411966
2021-01-06	555.000000	555.000000	541.000000	549.000000	53030554
2021-01-07	554.000000	570.000000	553.000000	565.000000	51166782
2021-01-08	580.000000	580.000000	571.000000	580.000000	59563555

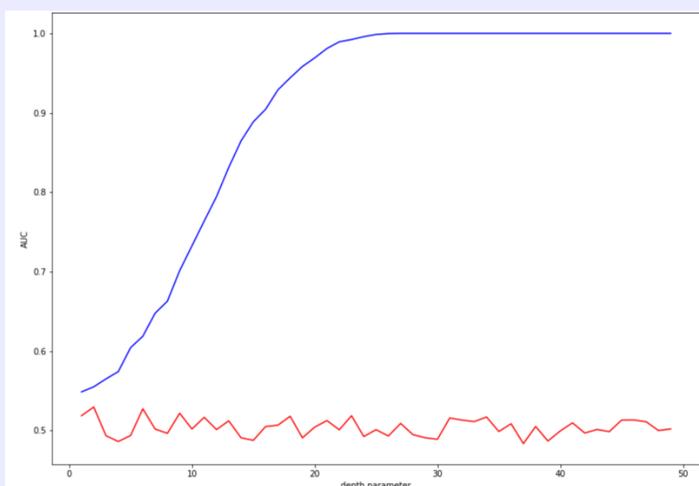
5180 rows x 5 columns



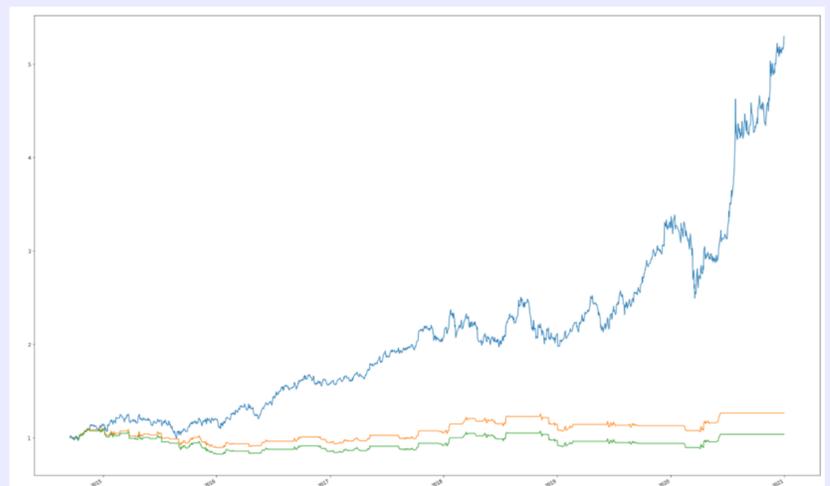
1. 從股市網站抓一支股票的資料

2. 預測五日後的漲跌(紅漲綠跌)

3. 匯出決策數



4. 計算1-50決策樹深度的混淆矩陣及AUC面積的準確率



5. 計算策略報酬及扣除成本報酬

結論

在建立這個程式碼時遇到了很多難題，雖然也沒到很完善，還有很多地方要改進加強，像是準確率一直無法提升，在45-60%之間，可能是城市無法完整預測突發狀況或者影響股價因素實在太多，就算是AI在這個階段也可能無法完全預測，畢竟如果能完全預測，這樣股市可能就不會有輸家在了。

資料來源

<https://medium.com/ai%E8%82%A1%E4%BB%94/%E5%AD%B8%E6%9C%83%E7%94%A8%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E9%A0%90%E6%B8%AC%E8%82%A1%E5%83%B9-%E5%AE%8C%E6%95%B4%E6%B5%81%E7%A8%8B%E6%95%99%E5%AD%B8%E8%88%87%E5%AF%A6%E4%BD%9C-b057e7343ca4>

義守大學 財務與計算數學系

AI+機器學習 專題成果

COVID-19的應用

組員：10640011A洪健豪、10640016A郭聿庭、10640044A莊琇晴

指導老師：羅大欽老師、葉建寧老師

摘要

2020 1月份從武漢爆發新冠病毒，到現在影響整個世界，大家都被這個影響，無法正常生活、工作、不管什麼事情都因疫情停擺。一開始沒有人把疫情看的很嚴重，後來越來越多人確診、死亡人數也逐漸上升，大家才把這個當很嚴重的事來看。所以我們要判斷這個是否會因民俗風情、醫療、政策和民眾看待疫情的嚴重性，使確診人數上升。我們使用python來判斷這9個國家的折線圖，確診結果是否源自於曾經參與重大活動，使得幾天後的感染人口上升。

專題問題描述

預測每個國家的疫情走向，並依預測結果判斷和醫療體系有無相關性。
分析預測結果中標率高的原因為何，是否和人口、政府、生活習慣...有無相關。

方法與過程

確診人數&死亡率 程式碼

```
import pandas as pd
import matplotlib.pyplot as plt

def abc(x):
    path = "Total.csv"
    data1 = pd.read_csv(path, delimiter=',', index_col='Date')
    datax=data1[data1['Entity'] == x]

    data2 = datax[['Daily new confirmed cases of COVID-19', 'Daily new confirmed deaths due to COVID-19']]
    data3 = data2.transpose()
    X=data3.columns
    plt.figure(figsize=(18, 10))

    plt.plot(list(X), data3.iloc[0,:]) # 確診人數
    plt.plot(list(X), data3.iloc[1,:]/data3.iloc[0,:]) # 死亡率

    plt.xticks(list(X)[:20])
    #print(data3)
A=input("選擇國家 : ")
abc(A)
```

選擇國家 :

確診人數 折線圖 程式碼

```
import pandas as pd
import matplotlib.pyplot as plt

country = ['United States', 'Brazil', 'Argentina', 'South Korea', 'Japan', 'Indonesia', 'Italy', 'United Kingdom', 'France']

def abc():
    path = "Total.csv"
    data1 = pd.read_csv(path, delimiter=',', index_col='Date')

    plt.figure(figsize=(18, 10))

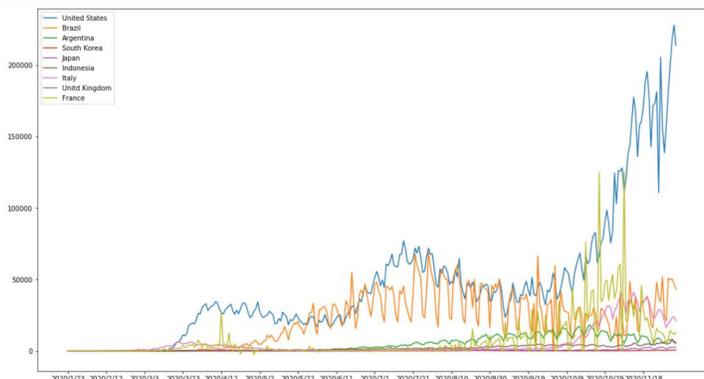
    for x in country:
        datax=data1[data1['Entity'] == x]

        data2 = datax[['Daily new confirmed cases of COVID-19', 'Daily new confirmed deaths due to COVID-19']]
        data3 = data2.transpose()
        X=data3.index

        plt.plot(list(X), data3.iloc[0,:], label=x)
        plt.legend()

    plt.xticks(list(X)[:20])
    #print(data3)

abc()
```



死亡率 折線圖 程式碼 (9個國家)

```
import pandas as pd
import matplotlib.pyplot as plt

country = ['United States', 'Argentina', 'Brazil', 'United Kingdom', 'Italy', 'France', 'Japan', 'Indonesia', 'South Korea']

def abc():
    path = "Total.csv"
    data1 = pd.read_csv(path, delimiter=',', index_col='Date')

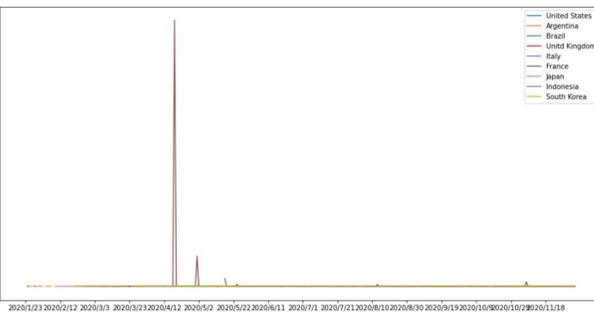
    plt.figure(figsize=(16, 8))

    for x in country:
        datax=data1[data1['Entity'] == x]

        data2 = datax[['Daily new confirmed cases of COVID-19', 'Daily new confirmed deaths due to COVID-19']]
        data3 = data2.transpose()
        X=data3.index

        plt.plot(list(X), data3.iloc[1,:]/data3.iloc[0,:], label=x)
        plt.legend()
    plt.xticks(list(X)[:20])

abc()
```



死亡率 折線圖 程式碼 (不包含法國)

```
import pandas as pd
import matplotlib.pyplot as plt

country = ['United States', 'Argentina', 'Brazil', 'United Kingdom', 'Italy', 'Japan', 'Indonesia', 'South Korea']

def abc():
    path = "Total.csv"
    data1 = pd.read_csv(path, delimiter=',', index_col='Date')

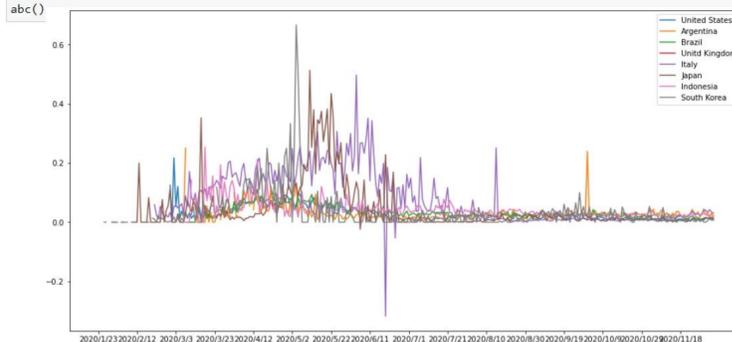
    plt.figure(figsize=(16, 8))

    for x in country:
        datax=data1[data1['Entity'] == x]

        data2 = datax[['Daily new confirmed cases of COVID-19', 'Daily new confirmed deaths due to COVID-19']]
        data3 = data2.transpose()
        X=data3.index

        plt.plot(list(X), data3.iloc[1,:]/data3.iloc[0,:], label=x)
        plt.legend()
    plt.xticks(list(X)[:20])

abc()
```



結論

最重要的是，沒有任何一種防疫措施是完美的。
最好的抗疫方式還是減少人與人間的接觸，並確保正確洗手與積極配合各項政府指示。

參考資料

- [1] <https://ourworldindata.org/>
- [2] <https://www.cna.com.tw/news/aopl/202010230067.aspx>
- [3] <https://news.campaign.yahoo.com.tw/lung-health/arti.php?id=c96b5eea-e54d-354e-8f19-7ae5f48b6966>
- [4] <https://opinion.cw.com.tw/blog/profile/436/article/9220>



義守大學 財務與計算數學系

AI+機器學習 專題成果

手寫數字辨識

組員：林帛翰 鍾沛辰 黃仕胤 許家瑞

指導教授：謝良瑜 老師

摘要

MNIST 是一個手寫資料數據集，其中包含著 60000 張 Training data 的圖片，以及 10000 張 Test data 的圖片，每張的像素皆為 28×28 ，每個像素點都以一個灰階值來表示。我們將使用這個數據集來訓練出一個可以分類手寫數字的模型，再用這個模型透過影像輸入的方式來辨識我們手寫的數字。

本次將使用 Jupyter Notebook Python 及 Keras 套件讀入 MNIST，把二維圖像轉換為 $n \times 28 \times 28 \times 1$ 的陣列並將數字標準化，再把標籤號轉換為 one hot encoding，利用 CNN 來訓練模型。再使用 OpenCV 開啟鏡頭讀取影像，並利用訓練好的模型來辨識圖片。

MNIST - Data

像素為 28×28 的灰階影像，影像資料由 28×28 的陣列來儲存，陣列裡的元素從 0~255 數字代表亮度，數字越大越亮。

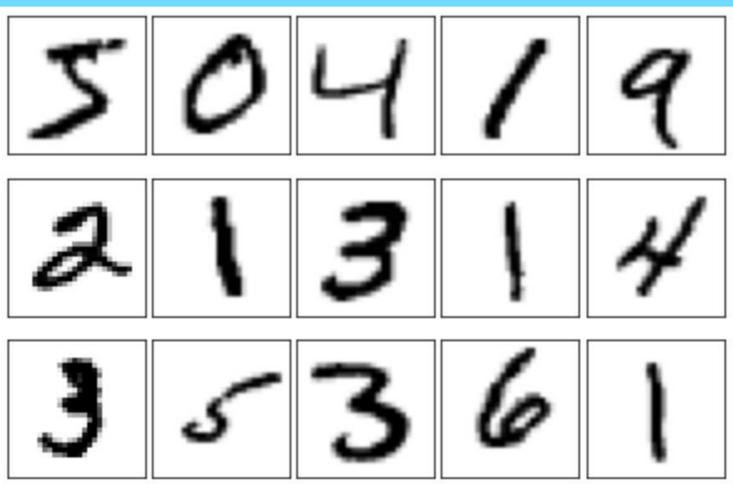


圖1. MNIST - Data 前15筆影像

MNIST - Label

label 為對應至 data 的數字資料

[5, 0, 4, 1, 9, 2, 1, 3, 1, 4, 3, 5, 3, 6, 1]

圖2. MNIST - Label 前15筆資料

Label - one-hot encoding

one-hot encoding 是一種有效編碼，用於表示名目尺度的一種編碼格式，使用時較有效率。

```
[0., 0., 0., 0., 0., 1., 0., 0., 0., 0.]
[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
[0., 0., 0., 0., 1., 0., 0., 0., 0., 0.]
[0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]
[0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]
```

圖3. Label - one-hot encoding 前5筆資料

成果

程式架構

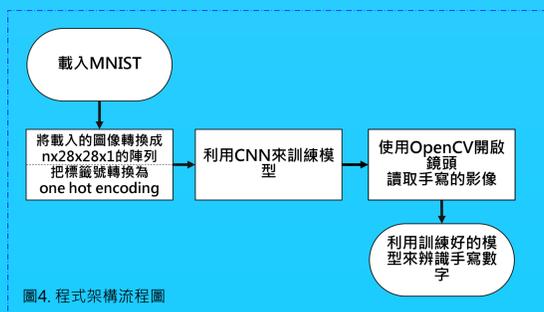


圖4. 程式架構流程圖

模型預測實作 1

在方形畫紙上手寫數字來做模型預測



圖5. 在小畫家上手寫數字 6



圖7. 在小畫家上手寫數字 5

```
#預測
model.predict_classes(X)
array([6], dtype=int64)
```

圖6. 圖5的預測結果為 6

```
#預測
model.predict_classes(X)
array([5], dtype=int64)
```

圖8. 圖7的預測結果為 5

模型預測實作 2

啟用電腦鏡頭或網路攝像頭來讀入資料並使用模型來做預測



圖8. ▲ 網路攝像頭讀入視窗 ▲ 模型預測結果視窗

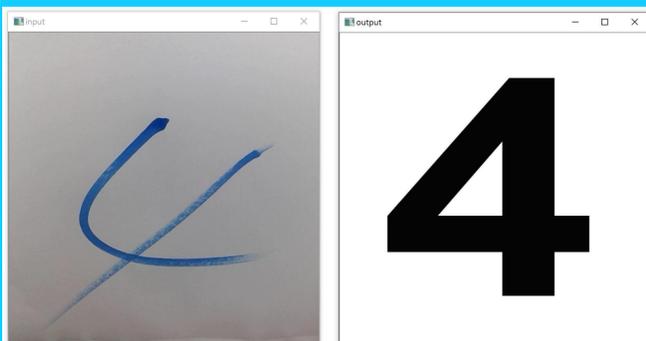


圖9. ▲ 網路攝像頭讀手寫數字 4 ▲ 模型預測結果為 4

程式碼簡易說明

程式碼主要分成6個部分，分別為載入手寫數字資料、資料轉換、模型訓練、模型測試、模型預測實作1、模型預測實作2

資料轉換是為了使用 CNN 模型及更有效率處理資料，所需要的前置作業。

模型訓練的部分使用的是 Sequential 是函數式模型的簡略版，為最簡單的線性從頭到尾的結構順序，不分叉，是多個網路層的線性堆疊。

模型內加入了兩層卷積層和池化層，就壓平放入 ANN，做兩層隱藏層後輸出

要執行模型預測實作需先在 Anaconda 裡安裝 Tensorflow、Keras、opencv 這三個套件，還需要一個鏡頭或網路攝像機

執行模型預測實作1，必須使用方形手寫數字圖片



圖5. 程式碼

結論

阿拉伯數字一共也就十個符號，特徵也比較好抓取，在整個模型訓練上也比較容易，僅做了兩層卷積層和池化層就放進類神經網路，神經網路裡也僅加了兩層隱藏層，就能使由 MNIST 裡訓練集所訓練出來的模型，用來預測測試集的準確度高達 99% 左右。

由此可見，訓練好的模型用來估計 MNIST 裡 28×28 的灰階手寫數字影像準確度是非常高的，但是在實際模型預測實作時準確度並不是非常的高，這說明了模型預測實作的部分，將方形影像轉換及將解析度降低為 28×28 時失真了，使影像原有的特徵改變，才導致模型精準度下降，這個部份我們還在設法優化。

參考文獻

- (1)使用 CNN 進行 MNIST 的手寫數字辨識 —— by Keras (實戰篇) <https://reurl.cc/0DzqGb>
- (2) Cinnamon AI Taiwan —— 深度學習：CNN原理 <https://reurl.cc/0DzqbY>
- (3)究竟神經網路是什麼？ —— 第一章 深度學習 <https://reurl.cc/mqd6aW>
- (4)高怡宣老師的個人網頁 —— AI-人工智慧 <https://reurl.cc/4yg5Ej>

義守大學 財務與計算數學系

AI+機器學習 專題成果

GOOGLE CLOUD PEOPLE DETECTION

組員：黃書涵 張簡暉哲
指導老師：郎正廉

摘要

先了解GOOGLE CLOUD是如何使用的，再利用GOOGLE CLOUD裡的人物檢測系統，分析人類的照片或視頻並嘗試定位其身體部位。

專題問題描述

利用機器學習分析影片，判斷網球選手動作是否錯誤，使用google cloud video Intelligence API得到影片數據。

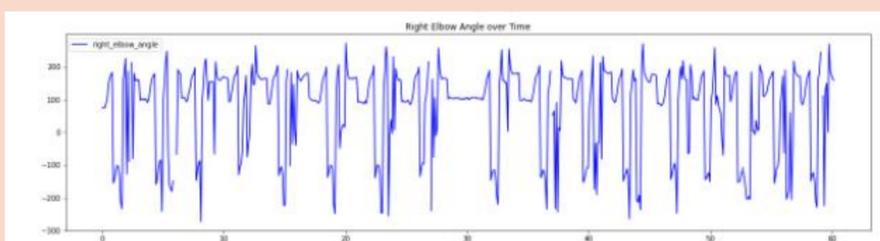
方法與過程

```
def detect_person(input_uri, output_uri):  
    """Detects people in a video."""  
  
    client = videointelligence.VideoIntelligenceServiceClient(credentials=service_account.Credentials.from_key('key.json'))  
  
    # Configure the request  
    config = videointelligence.types.PersonDetectionConfig(  
        include_bounding_boxes=True,  
        include_attributes=True,  
        include_pose_landmarks=True,  
    )  
    context = videointelligence.types.VideoContext(person_detection_config=config)  
  
    # Start the asynchronous request  
    operation = client.annotate_video(  
        input_uri=input_uri,  
        output_uri=output_uri,  
        features=[videointelligence.enums.Feature.PERSON_DETECTION],  
        video_context=context,  
    )  
  
    return operation
```

```
class Point:  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y  
  
    def getAngle(a, b, c):  
        ang = math.degrees(math.atan2(c.y-b.y, c.x-b.x) - math.atan2(a.y-b.y, a.x-b.x))  
        return ang  
  
    def computeElbowAngle(row, which='right'):  
        wrist = Point(row[f'{which}_wrist_x'], row[f'{which}_wrist_y'])  
        elbow = Point(row[f'{which}_elbow_x'], row[f'{which}_elbow_y'])  
        shoulder = Point(row[f'{which}_shoulder_x'], row[f'{which}_shoulder_y'])  
        return getAngle(wrist, elbow, shoulder)  
  
    def computeShoulderAngle(row, which='right'):  
        elbow = Point(row[f'{which}_elbow_x'], row[f'{which}_elbow_y'])  
        shoulder = Point(row[f'{which}_shoulder_x'], row[f'{which}_shoulder_y'])  
        hip = Point(row[f'{which}_hip_x'], row[f'{which}_hip_y'])  
        return getAngle(hip, shoulder, elbow)  
  
    def computeKneeAngle(row, which='right'):  
        hip = Point(row[f'{which}_hip_x'], row[f'{which}_hip_y'])  
        knee = Point(row[f'{which}_knee_x'], row[f'{which}_knee_y'])  
        ankle = Point(row[f'{which}_ankle_x'], row[f'{which}_ankle_y'])  
        return getAngle(ankle, knee, hip)
```

首先，我將網球發球的視頻剪輯到了我發球的部分，由於我只在相機上抓到17個發球區，因此花了我大約一分鐘的時間。接下來，我將視頻上傳到Google Cloud Storage，並通過Video Intelligence API對其進行了運行，所有這些代碼都方便地記錄在Colab筆記本中，你可以在自己的視頻中運行自己（只需要一個Google Cloud帳戶）。

使用這些數據，我可以非常準確地告訴我在什麼時候丟球並擊中球，我想將其與我擊球時肘部形成的角度對齊，為此，我必須將Video Intelligence API的輸出（原始像素位置）轉換為角度。



通過調整手腕的高度和肘部的角度，我能夠確定該角度約為120度（不是筆直！）。

結論

雖然使用Google Cloud做機器分析比較簡單，但還是需要看過別人寫的，才能知道在做什麼，而別人寫的也不一定能完全看懂，所以只能照著別人做的執行一次，只是可能還需要更多的時間，才能完全搞懂他在做什麼，目前並沒有很深入的了解。

參考資料

<https://daleonai.com/machine-learning-for-sports>

<https://cloud.google.com/video-intelligence/docs/people-detection>