

臺北市立大安高級工業職業學校
電子科
專題報告
學習成效監控系統
Learning effectiveness monitoring
system

組長:電子三乙 陳逢維
組員:電子三乙 陳聖介

指導老師: 張顯盛

摘要

學習是人類與其他動物與眾不同的地方，許多成就都是來自人類努力的結果。然而我們也是懶散的，一找到機會就休息、娛樂，也因此大部分的人都失敗了。為了防止許多學子走向墮落，就要制止他們頹廢，從而改變對讀書的看法。我們將用許多的圖片當範本，利用 opencv 及樹莓派的鏡頭錄下你目前的狀況，再用類神經網路與範本得知你有沒有在讀書。

這專題大部分是課堂上沒教到的知識，尤其是類神經網路；也由於我們對專題的不瞭解，製作專題的步驟一改再改。但這也很難避免，因為一個完全陌生的領域，一定要經過大量的網路資源及老師講解後，才能正確的運用。

Abstract

Learning is what sets humans apart from other animals. Many achievements are the result of human efforts. however, we are also lazy. When we find opportunities, we rest and entertain, so most people fail. In order to prevent many students from becoming degenerate, it is necessary to stop them from relaxing and change their views on reading. We will use many pictures as templates, use opencv and Raspberry Pi lenses to record your current situation, and then use neural-like networks and templates to know if you are reading.

Most of this topic is not taught in the classroom, especially neural network-like; because we do not understand the topic, the steps of making the topic have changed. But this is also difficult to avoid, because a completely unfamiliar field must be properly used after a lot of network resources and teacher explanations.

目錄

內容

摘要	2
Abstract.....	3
目錄	4
圖目錄	5
表目錄	6
第一章 前言	7
1-1 專題製作背景及目的.....	7
1-2 預期成果.....	7
第二章 理論探討	8
2-1 樹莓派.....	8
2-2 OPEN CV 簡介	9
2-3 類神經網路(深度學習)	10
2-3-1 簡介.....	10
2-3-2 歷史.....	11
2-3-3 原理.....	11
2-3-4 卷積神經網路簡介.....	12
2-4 開源軟體庫	13
2-4-1 tensorflow.....	13
2-4-2 keras	14
2-4-3 比較 tensorflow 與 keras	15
第 3 章 專題設計	16
3-1 系統架構圖.....	16
3-2 系統流程圖.....	17
3-3 甘特圖	18
第 4 章 專題成果	19
4-1 電腦成果.....	19
4-2 網站呈現結果.....	20
第 5 章 結論和建議.....	21
5-1 結論.....	21
5-2 問題與解決.....	21
5-2-1 Tffrecord 與 keras.....	21
5-2-2 樹莓派效能不足.....	22
5-3 建議.....	22
參考文獻.....	23
附錄	24
附錄一 設備清單	24
附錄二 材料清單	24
附錄三 程式碼.....	25
附錄四 成員簡介	31

圖目錄

圖 1 樹莓派 gpio 接腳.....	8
圖 2 樹莓派 2B.....	8
圖 3 opencv.....	9
圖 4 類神經網路架構.....	10
圖 5 類神經網路發展.....	11
圖 6 神經系統.....	11
圖 7 捲積神經網路.....	12
圖 8 池化層.....	12
圖 9 cnn 應用實例.....	12
圖 10 深度學習連接之各層.....	13
圖 11 tensorflow.....	13
圖 12 keras.....	14
圖 13 系統架構圖.....	16
圖 14 系統流程圖.....	17
圖 15 電腦成果.....	19
圖 16 電腦模擬結果.....	20
圖 17 tfrecord(失敗成果)(左).....	21
圖 18 keras(右).....	21
圖 19 樹莓派效能不足的系統顯示.....	22

表目錄

表 1 各檔案格式優劣.....	14
表 2 各開源軟體庫比較.....	15
表 3 甘特圖.....	18
表 4 設備清單.....	24
表 5 材料清單.....	24
表 6 陳逢維.....	31
表 7 陳聖介.....	32

第一章 前言

1-1 專題製作背景及目的

近年來因手機等行動通訊系統發達，許多學生深陷網路的泥沼中，使得學習成效不佳。或許他們曾經嘗試克制過，然而這並不能夠持久。

此裝置希望能用類神經網路的特性判斷在讀書時的各種狀況，例如打瞌睡，滑手機等。以方便自己了解讀書時的效率是否達到最高。進而督促自己朝更好的方向邁進。


1-2 預期成果

1. 類神經網路能正確判斷是否有在讀書。
2. 樹莓派在判斷沒有讀書一定次數後，會發出警示音。
3. 能開發手機 APP 紀錄先前的讀書狀況，且在一定時間內可以封鎖指定手機 APP

第二章 理論探討

2-1 樹莓派

樹莓派是基於 Linux 的單晶片電腦，由英國樹莓派基金會開發，目的是以低價硬體及自由軟體促進學校的基本電腦科學教育。樹莓派配備一枚博通 (Broadcom) 出產的 ARM 架構 700MHz BCM2835 處理器，256MB 記憶體 (B 型已升級到 512MB 記憶體)，使用 SD 卡當作儲存媒體，且擁有一個 Ethernet、兩個 USB 介面、以及 HDMI (支援聲音輸出) 和 RCA 端子輸出支援。樹莓派面積只有一張信用卡大小，體積大概是一個火柴盒大小。

 **Raspberry Pi 3 GPIO Header**


















Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I ² C)		DC Power 5v	04
05	GPIO03 (SCL1, I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

圖 1 樹莓派 gpio 接腳



圖 2 樹莓派 2B

2-2 open cv 簡介



圖 3 opencv

OpenCV 用 C++ 語言編寫，它的主要介面也是 C++ 語言，但是依然保留了大量的 C 語言介面。該庫也有大量的 Python, Java and MATLAB/OCTAVE (版本 2.5) 的介面。這些語言的 API 介面函式可以透過線上文件取得。現在也提供對於 C#, Ch Ruby 的支援。

所有新的開發和演算法都是用 C++ 介面。一個使用 CUDA 的 GPU 介面也於 2010 年 9 月開始實現

2-3 類神經網路(深度學習)

2-3-1 簡介

深度學習是指具有層次性的機器學習法，能透過層層處理將大量無序的訊號漸轉為有用的資訊並解決問題。但通常提到深度學習，人們指的是一種特定的機器學習法—「深度神經網路」(Deep Neural Network)。它包含許多神經元，有些負責接受資料，有些負責傳遞資料。例:cnn(捲積神經網路)這個神經網路包含三層神經元，除了輸入和輸出層外，中間有一層隱藏層(意指不參與輸入或輸出，隱藏於內部)，傳遞並處理資料。其實，隱藏層可以有一層以上，而複數個隱藏層的神經網路通常被稱為深度神經網路

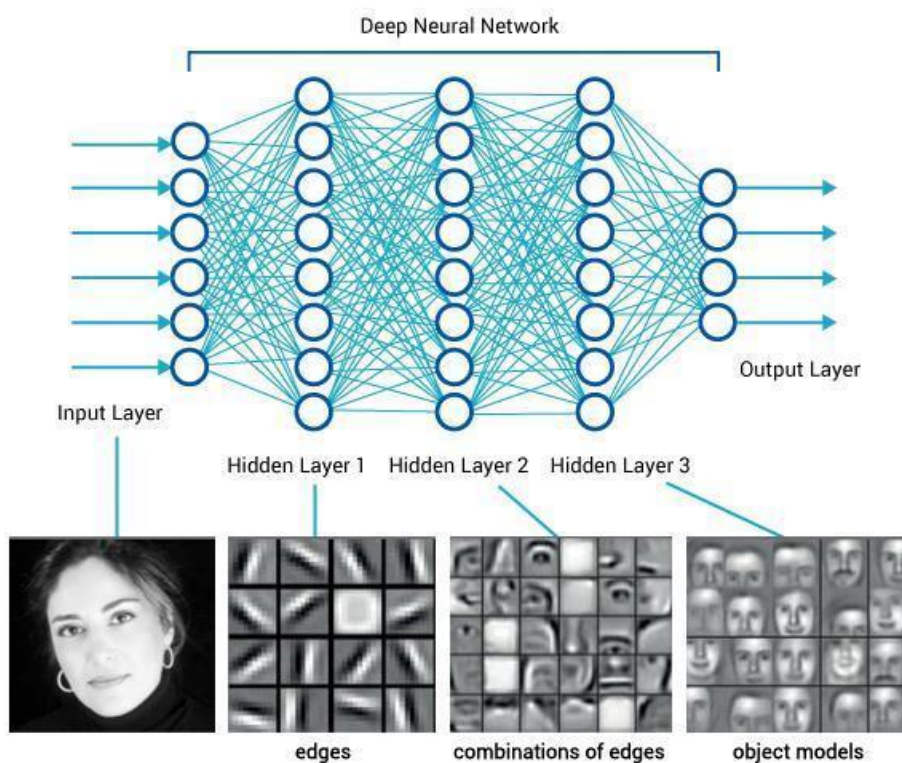


圖 4 類神經網路架構

2-3-2 歷史

1950 年代電腦發明以來，科學家便希冀著利用電腦創造出人工智慧；然而，當時的硬體效能低落、數據量不足，隨著通用問題解決機、日本第五代電腦等研究計畫的失敗，人工智慧陷入了第一次的寒冬。

1981 年，美國神經生物學家 David Hubel 和 Torsten Wiesel 對於動物視覺系統的處理信息方式有了進一步的發現，因而獲得了諾貝爾醫學獎。生物學上的神經元研究，啟發了 AI 領域關於「類神經網路」（或稱人工神經網路）的概念。神經系統由神經元構成，彼此間透過突觸以電流傳遞訊號。

1986 年，Rumelhar 和 Hinton 等人提出了反向傳播算法，解決了神經網路所需要的複雜計算量問題，從而帶動了神經網路的研究熱潮。30 年以後，Hinton 重新定義了神經網路；Hinton 也因此被稱為「深度學習之父」。

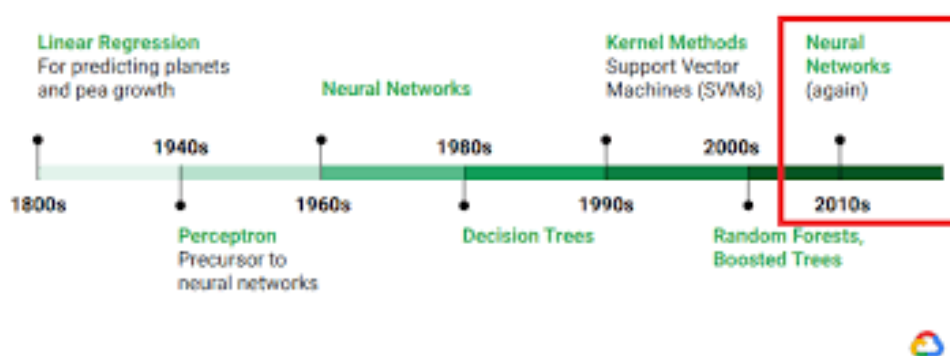


圖 5 類神經網路發展

2-3-3 原理

類神經網路模仿大腦運作的機器學習方式。人類大腦內的神經網路，主要是由神經元與突觸所組成。其中，神經元具有階層性，階層之間的神經元有著不同強度的鍵結，而類神經網路，就是仿照這樣的概念發展而成

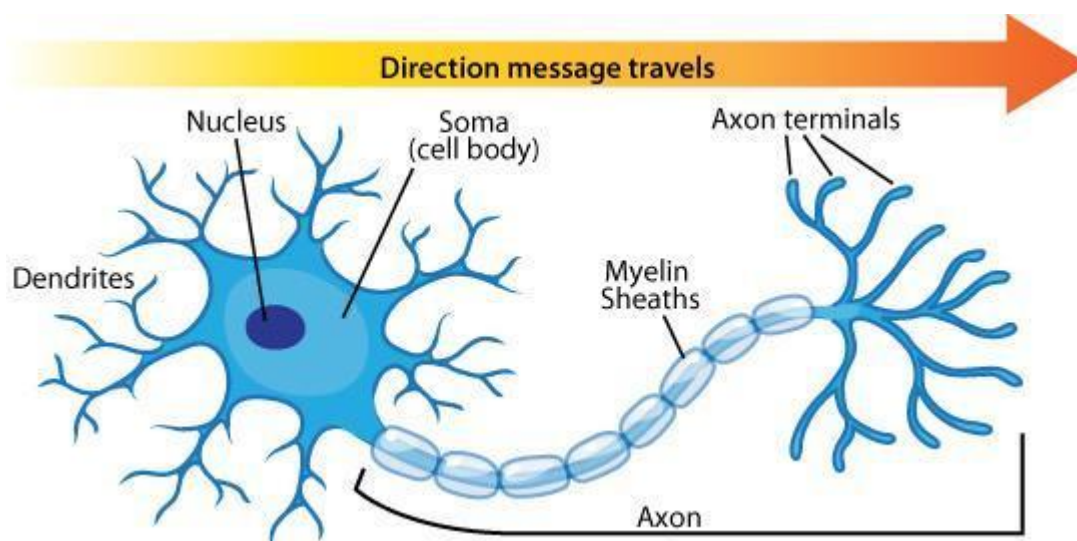


圖 6 神經系統

2-3-4 卷積神經網路簡介

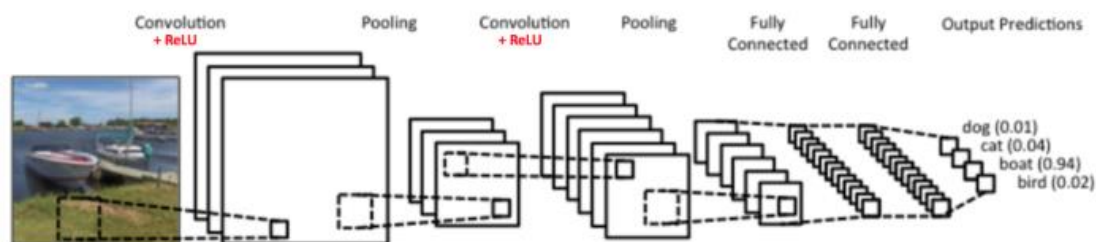


圖 7 捲積神經網路

卷積神經網路模仿人類大腦的認知方式，譬如我們辨識一個圖像，會先注意到顏色鮮明的點、線、面，之後將它們構成一個個不同的形狀(眼睛、鼻子、嘴巴...)，這種抽象化的過程就是 CNN 演算法建立模型的方式。卷積層(Convolution Layer)就是由點的比對轉成局部的比對，透過一塊塊的特徵研判，逐步堆疊綜合比對結果，就可以得到比較好的辨識結果。

卷積層(Convolution Layer)

以圖像的每一點為中心，取周遭 $N \times N$ 格的點構成一個面(N 稱為 Kernel Size, $N \times N$ 的矩陣權重稱為『卷積核』)，每一格給予不同的權重，計算加權總和，當作這一點的 output，再移動至下一點以相同方式處理，至圖像的最後一點為止。

池化層(Pooling Layer)

它是一個壓縮圖片並保留重要資訊的方法，取樣的方法一樣是採滑動視窗，但是通常取最大值(Max-Pooling)，而非加權總和。若滑動視窗大小設為 2，『滑動步長』(Stride) 也為 2，則資料量就降為原本的四分之一。

池化 最常見的方式有二種，最大池化(Max Pooling) 與平均池化(Average Pooling)。最大池主要針對影像紋理(Texture)，較常運用在影像識別、偵測方面；平均池化主要用於保留影像背景資訊，較常使用在風格轉換、映射方面。

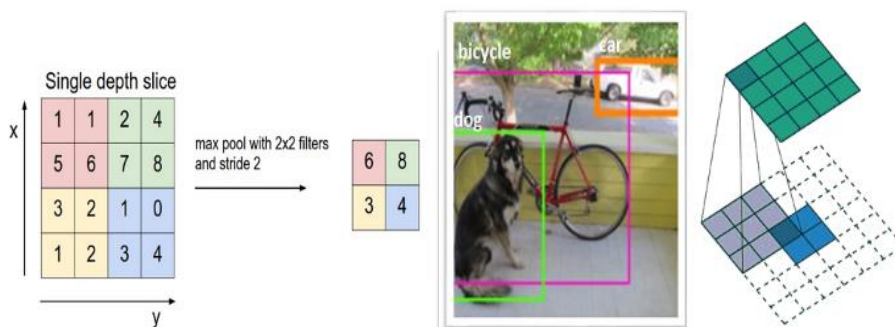


圖 8 池化層

圖 9 cnn 應用實例

透過多層卷積/池化，萃取特徵當作 Input，再接至一到多個完全連接層，進行分類，這就是 CNN 的典型作法。

2-4 開源軟體庫



圖 10 深度學習連接之各層

2-4-1 tensorflow



圖 11 tensorflow

Tensor=張量 Flow=流

TensorFlow 是一個用於機器學習的開源軟體庫，可以支援深度學習的各種演算法。它最初是由傑夫·迪恩 (Jeff Dean) 領軍的 Google Brain 團隊，基於 Google 第一代深度學習系統 DistBelief 改進而產生。

總體來說 TensorFlow 就是為了快而設計的，它針對你實際使用的硬體和平臺做了優化。其中在機器學習框架方面，TensorFlow 能夠在 5 行或者 10 行程式碼中構建模型，進行擴展做出產品。事實上對於類似電腦視覺的問題，過去的神經網路幾乎無用武之地。相比於為某個確切的任務，如今可以用來與圖像分類的神經網路作比較，這個被稱為 Inception。它的設計思路是給出一張圖片，它就能從成千上萬，大約 1 萬 7 千種類別中，預測出圖中包含了什麼。你可以很輕鬆地實現反向傳播。

TFRecord 是一種專為 Tensorflow 打造的儲存文件的格式。

表 1 各檔案格式優劣

格式	儲存速度	讀取速度	檔案大小
.npy	最快	最快	最小
.pkl	中	最慢	最大
.tfrecord	最慢	中	中

2-4-2 keras



圖 12 keras

Keras 是一個開放原始碼，基於 Python 高階深度學習的程式庫，主要由 Francois Chollet 及其他開放原始碼社群成員一同開發，以 MIT 開放原始碼授權。Keras 可以快速有方便運算的主要原因是，它已經將訓練模型的輸入層、隱藏層、輸出層，做好架構，使用者只需要加入並且填寫正確的參數 ex. 神經元個數、activation function 的函式... 等。

Keras 是一個模型等級(Model-level)的程式庫，提供開發者所需要的高階模組，但 Keras 只處理模型的建立、訓練、預測等功能，因此它無法處理如張量和矩陣運算以及微分等底層作業，需要依賴特製且最佳化過的張量程式庫，作為 Keras 的後端引擎 (Backend Engine) 以進行張量運算等底層作業。目前 Keras 提供三種後端引擎分別是：TensorFlow、Theano 及 Microsoft 運算神經網路工具組(The Microsoft Cognitive Toolkit, CNTK)，相信 Keras 未來將擴展到更多深度學習的執行引擎。

2-4-3 比較 tensorflow 與 keras

表 2 各開源軟體庫比較

Library	Rank	Overall	Github	Stack Overflow	Google Results
tensorflow	1	10.87	4.25	4.37	2.24
keras	2	1.93	0.61	0.83	0.48
caffe	3	1.86	1.00	0.30	0.55
theano	4	0.76	-0.16	0.36	0.55
pytorch	5	0.48	-0.20	-0.30	0.98
sonnet	6	0.43	-0.33	-0.36	1.12
mxnet	7	0.10	0.12	-0.31	0.28
torch	8	0.01	-0.15	-0.01	0.17
cntk	9	-0.02	0.10	-0.28	0.17
dlib	10	-0.60	-0.40	-0.22	0.02

但是 keras 本身就是 tf、theano、cntk 的高層 API，所以把它作為一個獨立框架去與 tf、mxnet、pytorch 這些框架直接對比是不公平的，換句話說它的性能最根源還是決定於它的後端框架。

Tensorflow

優點: Tensorflow 更靈活，模組自訂程度更高。

缺點: 龐大的代碼文件庫，如果之前沒有程式設計經驗是比較慢上手的

Keras

優點: 可以很快地實現網路模型的搭建，資料登錄輸出也很方便，不像 tf 那樣各種編碼解碼才能輸入資料，容易上手

缺點: 作為高層 API，慢是必然的了，推理速度等等肯定沒 tf，mxnet 那些那麼快。

第 3 章 專題設計

3-1 系統架構圖

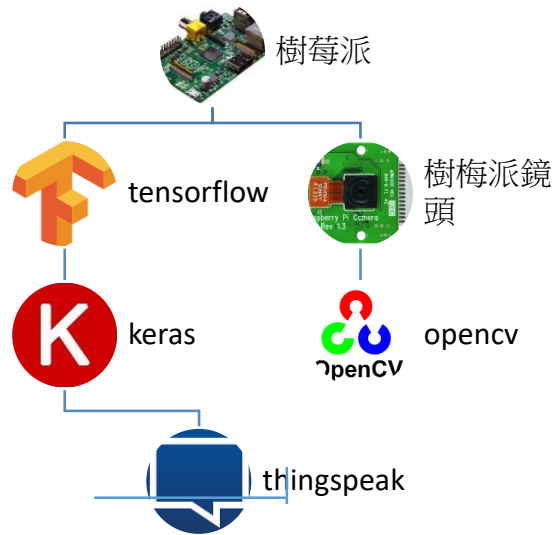


圖 13 系統架構圖

3-2 系統流程圖

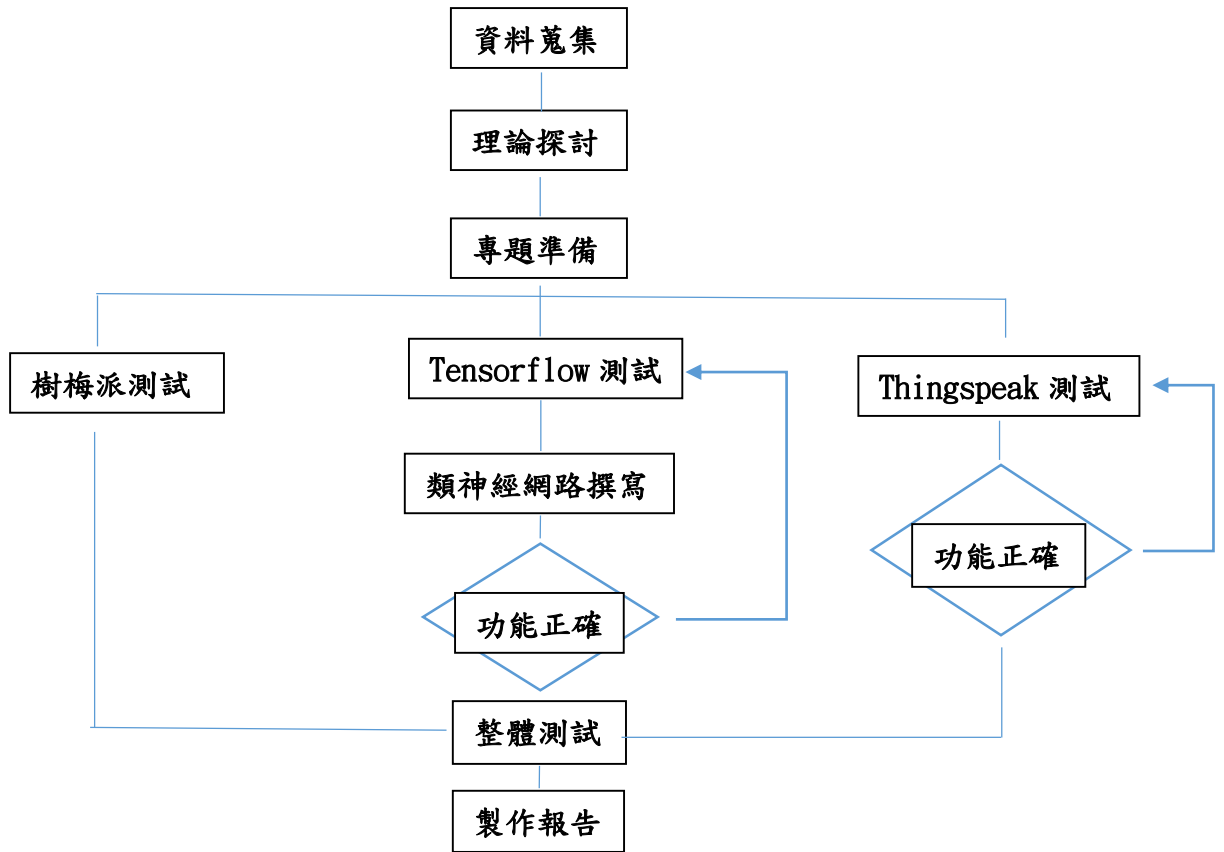


圖 14 系統流程圖

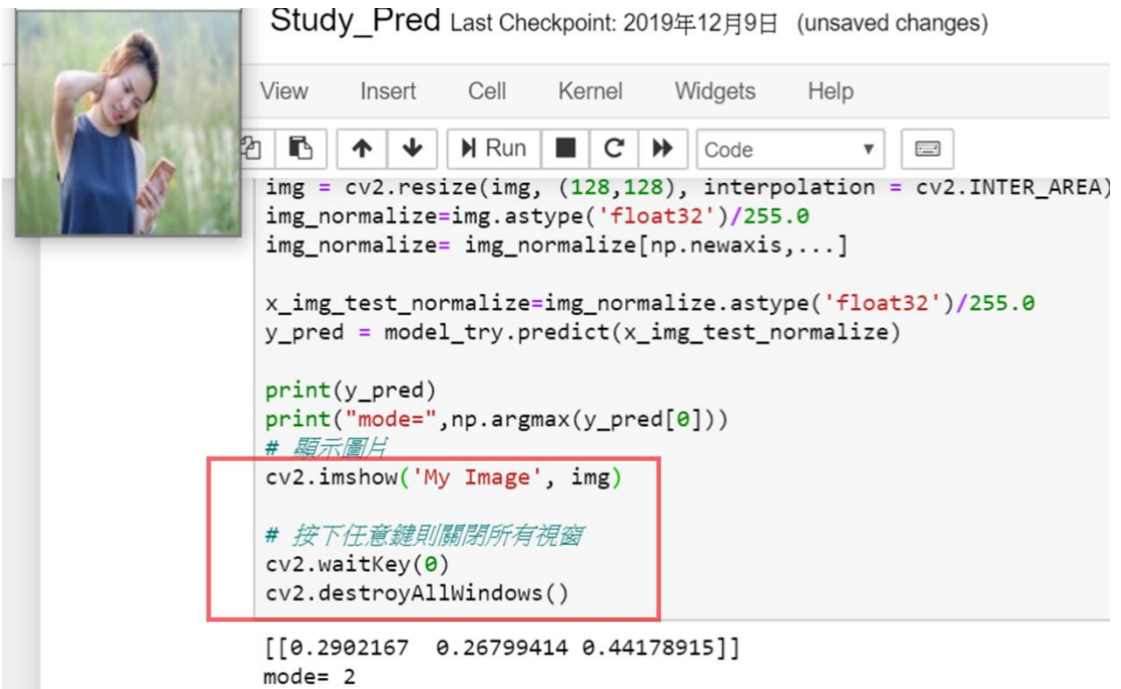
3-3 甘特圖

表 3 甘特圖

週次 (日期)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	負責成員	
工作項目																						
資料蒐集	■	■	■	■	■	■	■															all
理論探討	■	■	■	■	■	■	■	■							■							all
專題準備	■	■	■	■	■								■	■	■							All
樹梅派測試				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		30
Tensorflow 測試				■	■	■	■	■														27
opencv測試													■	■	■	■	■	■	■			27
類神經網路撰寫				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				30
各部整合																■	■	■	■			all
整體測試																		■	■	■	■	all
撰寫報告					■				■		■		■		■			■		■	■	all
口頭報告						■				■		■		■				■		■	■	all
預定進度	5	8	10	15	20	25	30	33	40	44	47	51	56	63	66	72	80	86	94	100	累積百分比(%)	

第 4 章 專題成果

4-1 電腦成果



```
Study_Pred Last Checkpoint: 2019年12月9日 (unsaved changes)
View Insert Cell Kernel Widgets Help
Run Code
img = cv2.resize(img, (128,128), interpolation = cv2.INTER_AREA)
img_normalize=img.astype('float32')/255.0
img_normalize= img_normalize[np.newaxis,...]

x_img_test_normalize=img_normalize.astype('float32')/255.0
y_pred = model_try.predict(x_img_test_normalize)

print(y_pred)
print("mode=", np.argmax(y_pred[0]))
# 顯示圖片
cv2.imshow('My Image', img)

# 按下任意鍵則關閉所有視窗
cv2.waitKey(0)
cv2.destroyAllWindows()

[[0.2902167  0.26799414  0.44178915]]
mode= 2
```

圖 15 電腦成果

4-2 網站呈現結果

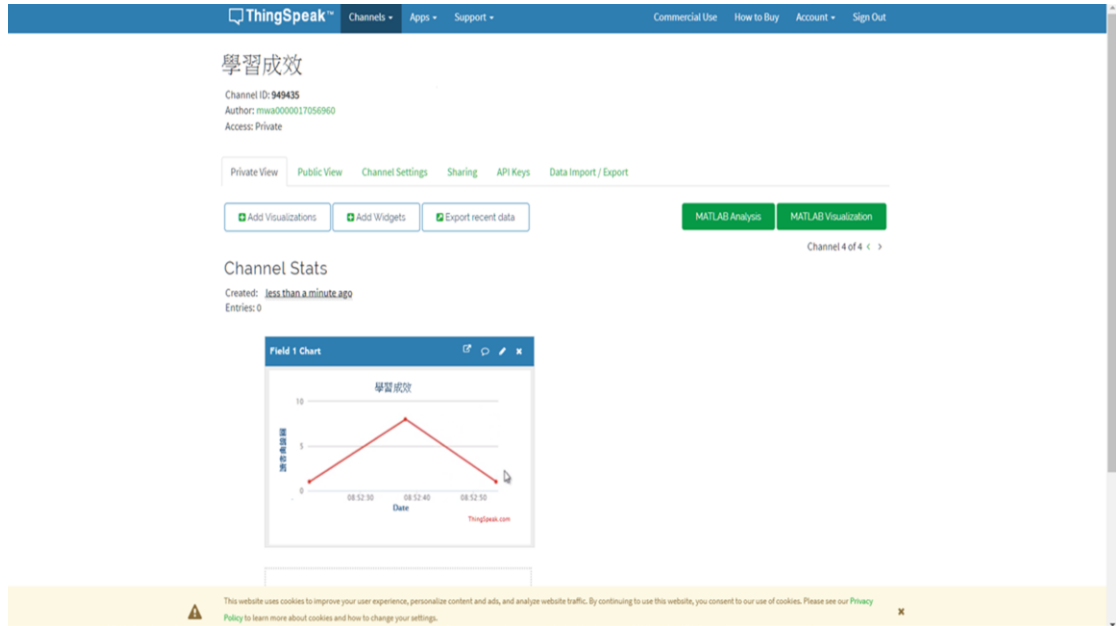


圖 16 電腦模擬結果

第 5 章 結論和建議

5-1 結論

由於我們這組人數太少，只有兩個人，沒有能力做正常電子科的專題題目，導致暑假都在討論題目，直到開學才真正開始做專題。然而這專題完全跟以前的課程沒有交集，這就表示我們要先瞭解樹莓派、類神經網路、及 opencv。所以我們找了許多資料，加上指導老師的協助，完成專題。

如果仔細看看類神經網路原理，就知道超出高中所能達到的知識水平了，但其實只要知道它的製作方法就好了，畢竟現在做專題的我們，都站在巨人的肩膀上，細節如何高深，要在花好久時間才能解答。

5-2 問題與解決

5-2-1 Tfrecord 與 keras

```
Reloaded modules: studying3, photoread2
Reading checkpoints...
INFO:tensorflow:Restoring parameters from D:/
train_data/photos\studying.ckpt-199
Loading success, global_step is 199
This is sleep with possibility 0.976838
```

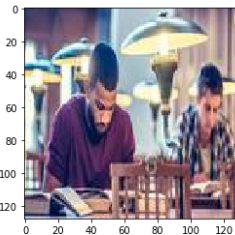


圖 17 tfrecord(失敗成果)(左)

```
Study_Pred Last Checkpoint: 2019年12月9日 (unsaved changes)
View Insert Cell Kernel Widgets Help
img = cv2.resize(img, (128,128), interpolation = cv2.INTER_AREA)
img_normalize=img.astype('float32')/255.0
img_normalize= img_normalize[np.newaxis,...]
x_img_test_normalize=img_normalize.astype('float32')/255.0
y_pred = model_try.predict(x_img_test_normalize)
print(y_pred)
print("mode=", np.argmax(y_pred[0]))
# 顯示圖片
cv2.imshow('My Image', img)
# 按下任意鍵則關閉所有視窗
cv2.waitKey(0)
cv2.destroyAllWindows()
[[0.2902167 0.26799414 0.44178915]]
mode= 2
```

圖 18 keras(右)

依照上面的圖片，使用 tfrecord 做的成功率太差了，甚至圖文不符，就算結果與內容相同恐怕也只是剛好猜對，而用 keras 做得成功率明顯增加。可能因為剛踏入影像辨識沒多久，不熟悉製作流程，且 keras 比較高階，相對而言比較人性化一點。另外也可能是查到的資料都是網路上的，發佈時間也幾年了，早就沒辦法配合目前版本，又或者作者多加了一些模組，而造成錯誤。

5-2-2 樹莓派效能不足

因為要建構神經網路需要非常繁雜的計算，然而樹梅派的記憶體無法負荷如此龐大的計算。因此專題最後的結果無法完成。

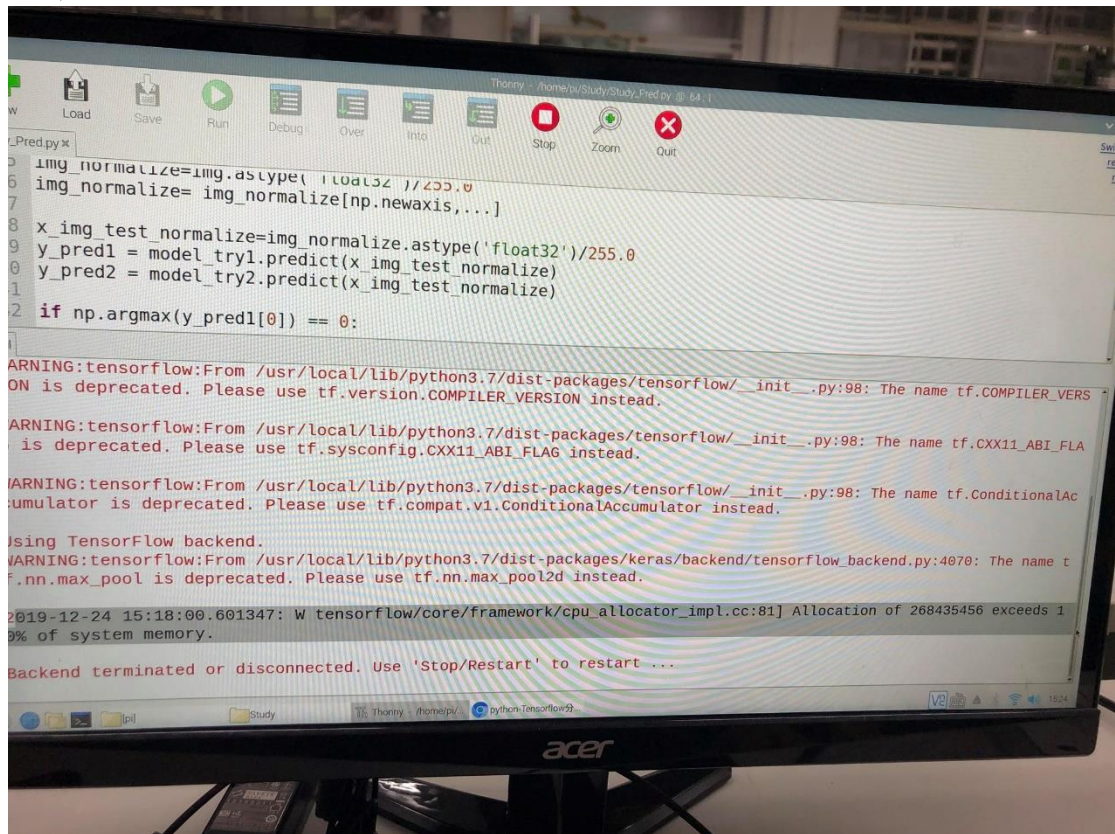


圖 19 樹莓派效能不足的系统顯示

5-3 建議

首先，要提早立方向(不管在何時都適用)。不管是幾人小組，一定要保持良好互動。通常這階段會消耗最長時間(在自訂題目下)。

要如何找資料也是一門學問，在找辨識圖片程式的資料時，有用 tfrecord 和 keras 兩種範例(keras 比 tfrecord 簡單)，起初我用 keras 寫的。但當時認為 keras 很難，就換成 tfrecord 的形式，而耽誤半個月。但失敗為成功之母，累積失敗的教訓就會改進。

參考文獻

類神經網路歷史:<https://www.stockfeel.com.tw> 機器學習的衰頹興盛：從類神經網路到淺層學習

卷積神經網路簡介:<https://ithelp.ithome.com.tw> Day 06：處理影像的利器 -- 卷積神經網路(Convolutional Neural Network)

Tensorflow:前: <https://www.techbang.com> 不懂程式你也該知道 Google 的 TensorFlow 究竟是什麼，以及它為何可能影響未來手機的硬體發展

後:<https://zhuanlan.zhihu.com> 谷歌大神帶你十分钟看懂 TensorFlow

Tfrecord:<https://ithelp.ithome.com.tw> TFRecord 在 Python 的使用

Keras 及池化:<https://tanet2019.nsysu.edu.tw> Keras 在 CIFAR-10 資料集之影像辨識分析

附錄

附錄一 設備清單

表 4 設備清單

類別	設備、軟體名稱	應用說明
硬體	電腦	撰寫程式
硬體	手機	APP 控制
軟體	anaconda	編寫程式

附錄二 材料清單

表 5 材料清單

類別名稱	材料名稱	單位	數量	應用說明	備註
硬體	樹莓派	個	1	媒介	
硬體	Camera Module v2	個	1	媒介	

附錄三 程式碼

```
import tensorflow as tf
import keras
import numpy as np
import os
import cv2
import matplotlib.pyplot as plt
np.random.seed(10)

label_dict={0:'studying',1:'sleep',2:'surf the phone' }
# def create_train_record(train_record_path,cwd):
work_path="photos"
work_num=0
x_img_train=np.array([[[[[]]]]])
x_img_train=x_img_train.reshape((0,128,128,3))
y_label_train=np.array([[[]]])
y_label_train=y_label_train.reshape((0,1))

lst_index=np.zeros((1,1),dtype=np.int)
# 讀取檔案進行訓練(128,128,3)
for index in range(len(label_dict)):
    class_path = work_path + "/" + label_dict[index] + '/'
    # lp = int(len(os.listdir(class_path)) * 0.7)      # 取前 70%建立訓練集
    lp = int(len(os.listdir(class_path)) * 1.0)      # 取前 70%建立訓練集
```

```

lst_index[0][0]=index
for img_name in os.listdir(class_path)[:lp]:
    img_path = class_path + img_name
    img = cv2.imread(img_path)
    img = cv2.resize(img, (128,128), interpolation =
cv2.INTER_AREA)

    img_normalize=img.astype('float32')/255.0
    img_normalize= img_normalize[np.newaxis,...]
    # print(y_label_train.shape, lst_index.shape)
    x_img_train = np.append(x_img_train, img_normalize, axis = 0)
    y_label_train = np.append(y_label_train, lst_index, axis = 0)
    work_num += 1

    #print("x_img_train.shape=", x_img_train.shape, "y_label_train.shape="
, y_label_train.shape)
    print(y_label_train[140:160])

def
plot_images_labels_prediction(images, labels, prediction, idx, num=10):
    fig=plt.gcf()
    fig.set_size_inches(12, 14)
    if num>25:num=25
    for i in range(0, num):
        ax=plt.subplot(5, 5, i+1)
        ax.imshow(images[idx], cmap='binary' )
        title=str(i) + "." + label_dict[labels[idx][0]]

```

```

    if len(prediction)>0:
        title+= "=>" + label_dict[prediction[idx]]

    ax.set_title(title, fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])
    idx+=1
plt.show()

#以樣本數量生成亂序的 list
permutation = list(np.random.permutation(x_img_train.shape[0]))
#按照隨機生成的順序重新排列資料集
x_x = x_img_train[permutation, :, :, :]
#訓練集有 4 個維度，第一個維度指的是樣本數量
y_y = y_label_train[permutation, :]
plot_images_labels_prediction(x_x, y_y, [], 52)

from keras.utils import np_utils
x_img_train_normalize=x_img_train.astype('float32')/255.0

y_label_train_OneHot = np_utils.to_categorical(y_label_train)

# 開始建立模型
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D
model = Sequential()#建立 keras 的 Sequential 模型

```

```

#建立卷積層 1
model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(128, 128, 3
), activation='relu', padding='same'))
#加入 Drop 層 1，避免 overfitting
model.add(Dropout(rate=0.25))
#建立池化層 1
model.add(MaxPooling2D(pool_size=(2, 2)))
#建立卷積層 2
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padd
ing='same'))
#加入 Drop 層 1，避免 overfitting
model.add(Dropout(rate=0.25))
#建立池化層 2
model.add(MaxPooling2D(pool_size=(2, 2)))
#神經網路開始
#建立平坦層
model.add(Flatten())
model.add(Dropout(rate=0.25))
#建立隱藏層
model.add(Dense(1024, activation='relu'))
model.add(Dropout(rate=0.25))
#建立輸出層
model.add(Dense(3, activation='softmax'))
print(model.summary())

#定義訓練方式
model.compile(loss='categorical_crossentropy', optimizer='adam', metri
cs=['accuracy'])

```

```

#以樣本數量生成亂序的 list
permutation =
list(np.random.permutation(x_img_train_normalize.shape[0]))
#按照隨機生成的順序重新排列資料集
x_img_train_normalize = x_img_train_normalize[permutation, :, :, :]
#訓練集有 4 個維度，第一個維度指的是樣本數量
y_label_train_OneHot = y_label_train_OneHot[permutation, :]
#開始訓練
train_history=model.fit(x_img_train_normalize,y_label_train_OneHot,v
alidation_split=0, epochs=50, batch_size=83, verbose=1, shuffle=True)

#scores=model.evaluate(x_img_test_normalize,y_label_test_OneHot,verb
ose=0)
scores=model.evaluate(x_img_train_normalize,y_label_train_OneHot,ver
bose=1)
print(" score=", scores[1])

model.save("Study_model.h5")
model.save_weights("Study_weight.h5")

#model.load_weights("StudyTrain.h5")
# del model

print(y_label_train_OneHot[10:30])

from keras.models import load_model

```

```

model_try = load_model('Study_model.h5')
model_try.load_weights("Study_weight.h5")

import cv2

# predict(x, batch_size=None, verbose=0, steps=None, callbacks=None,
max_queue_size=10, workers=1, use_multiprocessing=False)
# read file --> adjust size

#cv2 讀取檔案
img = cv2.imread('Study_2.jpg')
img = cv2.resize(img, (128,128), interpolation = cv2.INTER_AREA)
img_normalize=img.astype('float32')/255.0
img_normalize= img_normalize[np.newaxis,...]

x_img_test_normalize=img_normalize.astype('float32')/255.0
y_pred = model_try.predict(x_img_test_normalize)

print(y_pred)
print("mode=", np.argmax(y_pred[0]))

get_ipython().system(' jupyter nbconvert --to python
Study_Train.ipynb')

```

附錄四 成員簡介

表 6 陳逢維

姓名	陳逢維	班級	電子三乙	照 片
曾修習專業科目	1.基本電學與實習 2.電子學與實習 3.數位邏輯與實習 4.電子電路與實習 5.微處理機與實習			
參與專題工作項目	資料蒐集 理論探討 專題準備 Tensorflow 測試 樹梅派測試 手機 APP 測試 類神經網路撰寫 手機程式撰寫 整體測試 撰寫報告 口頭報告			
經歷簡介	工業電子丙級技術士證照 106 學年上學期擔任工廠領班 106 學年下學期擔任學藝幹事			

表 7 陳聖介

姓名	陳聖介	班級	電子三乙	
曾修習專業科目	1.基本電學與實習 2.電子學與實習 3.數位邏輯與實習 4.電子電路與實習 5.微處理機與實習			照片
參與專題工作項目	資料蒐集 理論探討 專題準備 Tensorflow 測試 樹梅派測試 手機 APP 測試 類神經網路撰寫 手機程式撰寫 整體測試 撰寫報告 口頭報告			
經歷簡介	工業電子丙級技術士證照 擔任圖書幹事(107)			