

專題成果發表

學習成效監測系統

Learning effectiveness monitoring system

組長:陳逢維
組員:陳聖介

指導老師:張顯盛

目錄

1

前言

2

專題流程

3

理論探討

4

遭遇問題

5

成果

6

未來期許

Section 1

前言

背景

4

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。

目的

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。

Section 2

專題流程

甘特圖

7

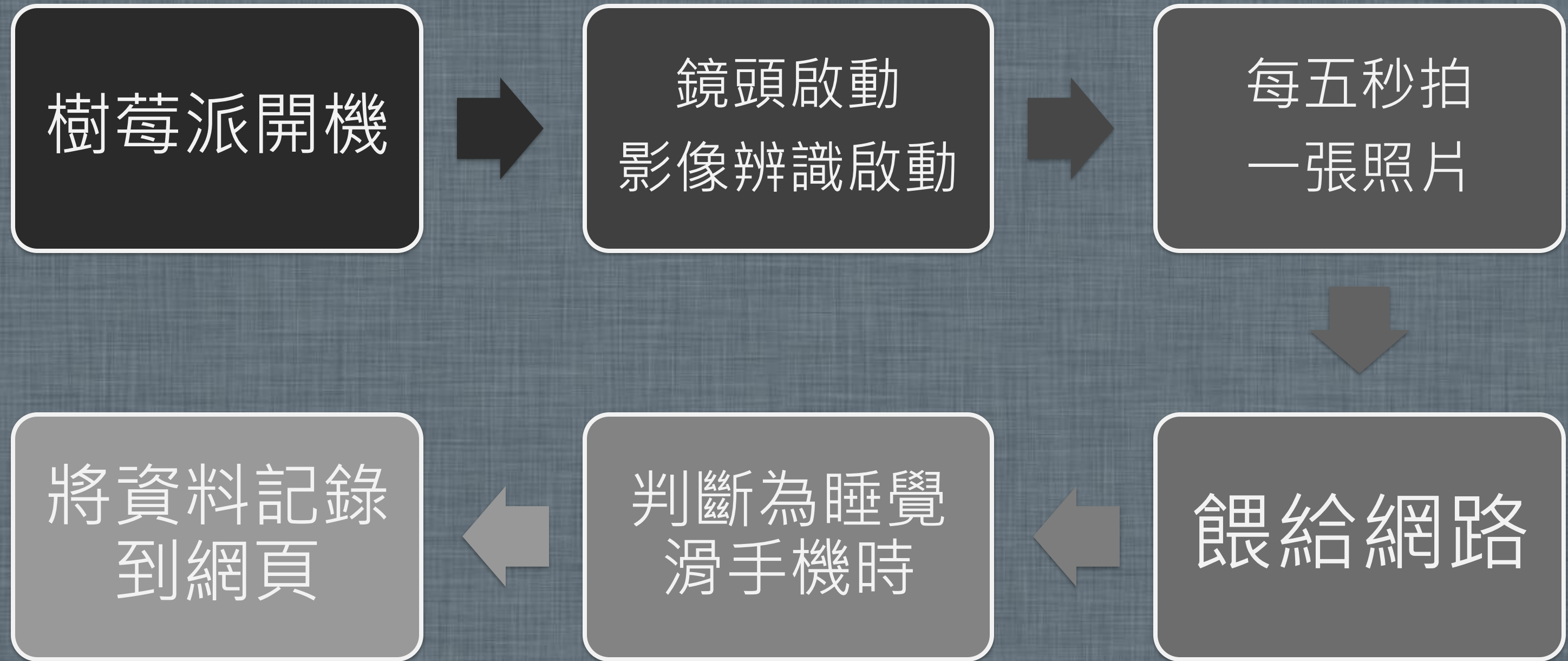
週次 (日期) 工作項目	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	負責成員	
資料蒐集	■	■	■	■	■	■	■														all	
理論探討	■	■	■	■	■	■	■	■							■							all
專題準備	■	■	■	■	■								■	■	■							all
樹梅派測試				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		27
Tensorflow 測試				■	■	■	■															30
opencv 測試												■	■	■	■	■	■	■				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(%)	

專題製作流程



作業流程

9

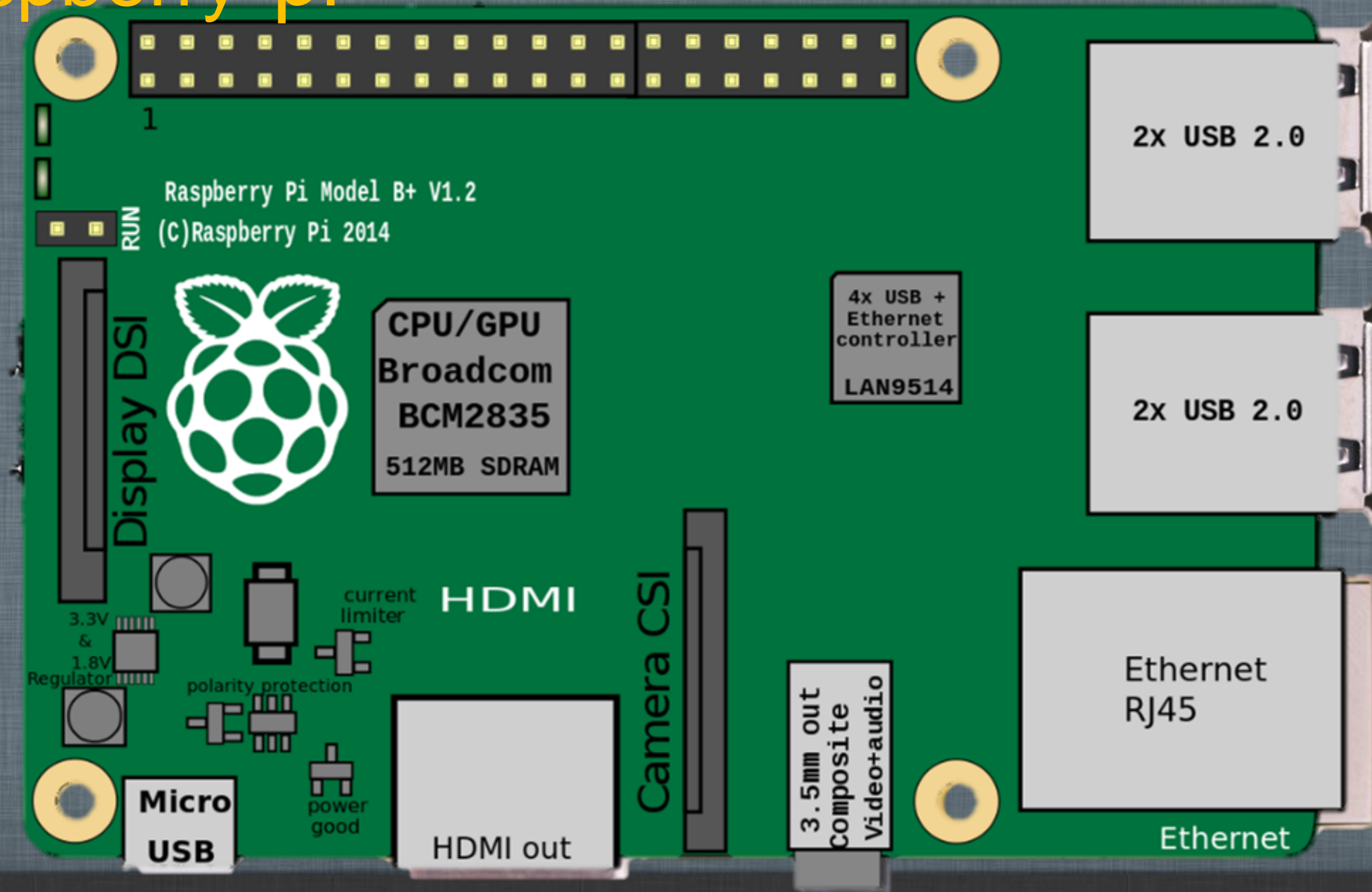


Section 3

理論研究

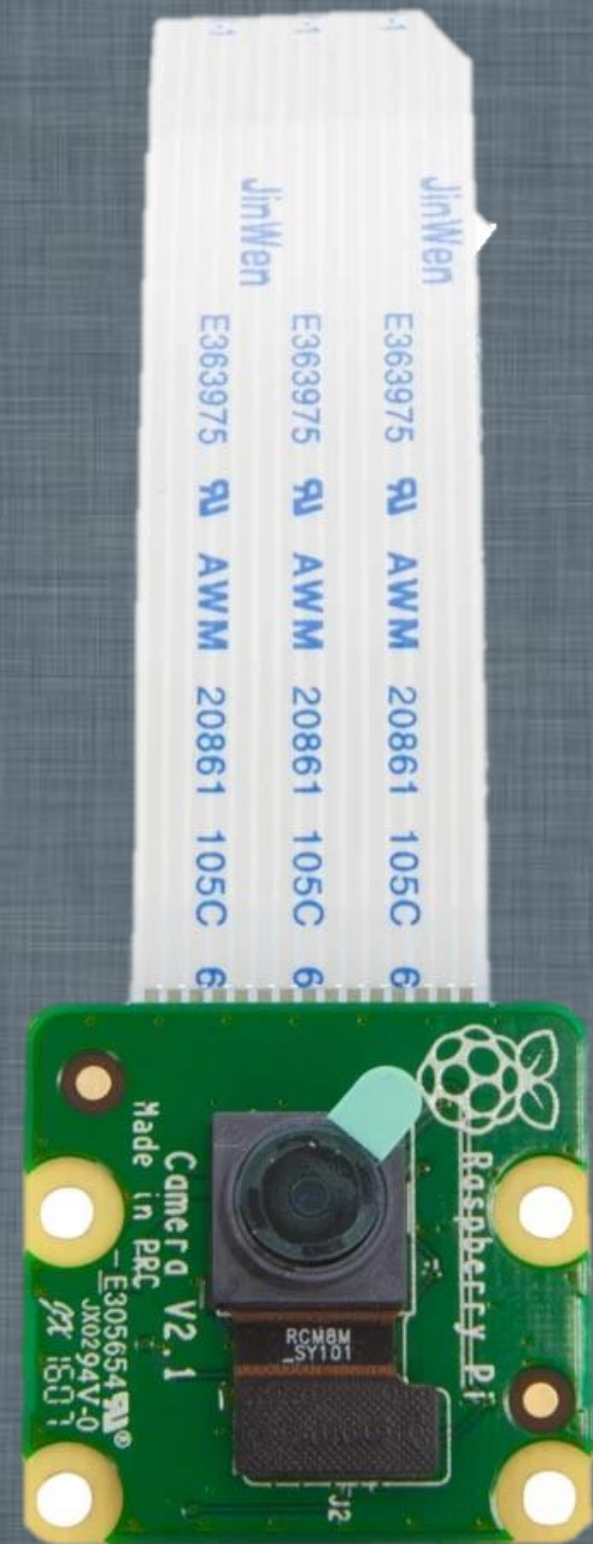
使用機構

Raspberry pi



相機

- 第二代覆盆子 Pi 相機模組，固定焦距鏡頭
- Sony Exmor IMX219 感應器 4K30、1080P60、720P180、8MP Still
- 3280 (H) x 2464 (V) 活化像素計數
- Raspberry Pi Board 最大 1080P30 和 8MP Stills



使用軟體

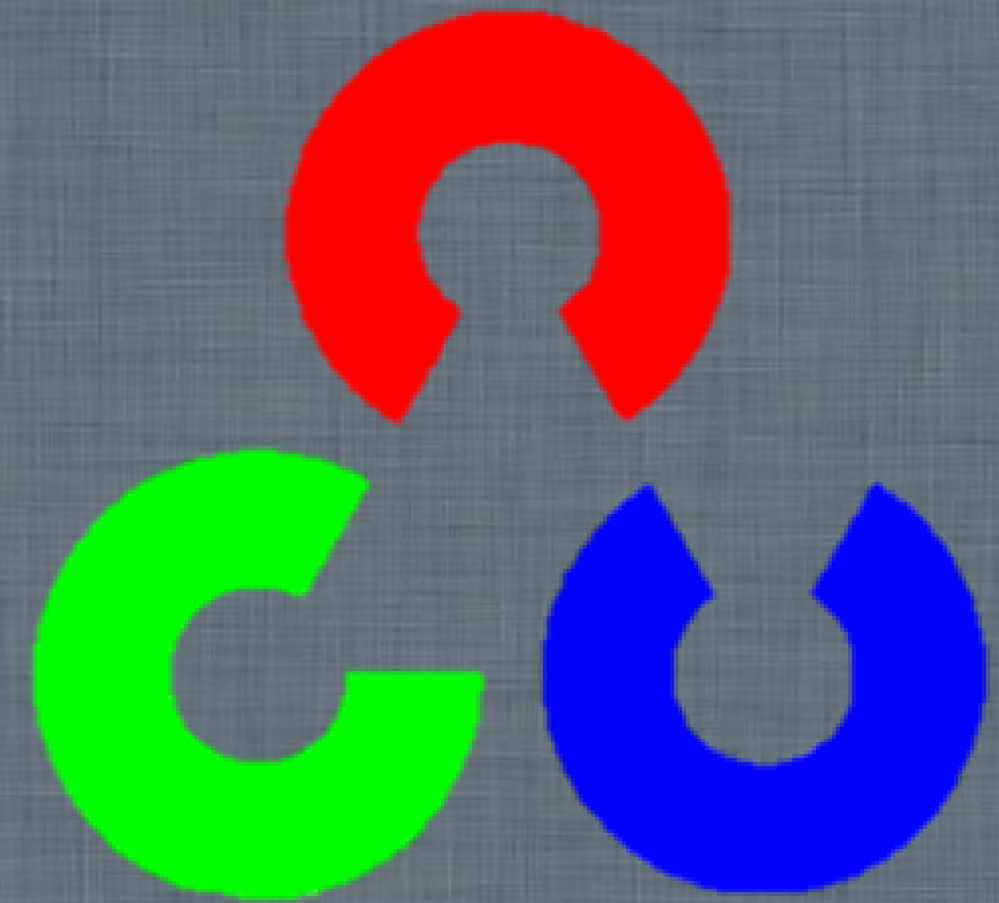
Opencv

OpenCV的全稱是Open Source Computer Vision Library，是一個跨平台的電腦視覺庫

可以執行在Linux、Windows、Android和Mac OS作業系統上

OpenCV可用於解決如下領域的問題：

擴增實境
臉部辨識
手勢辨識
人機互動
動作辨識
運動跟蹤
物體辨識
圖像分割
機器人



OpenCV

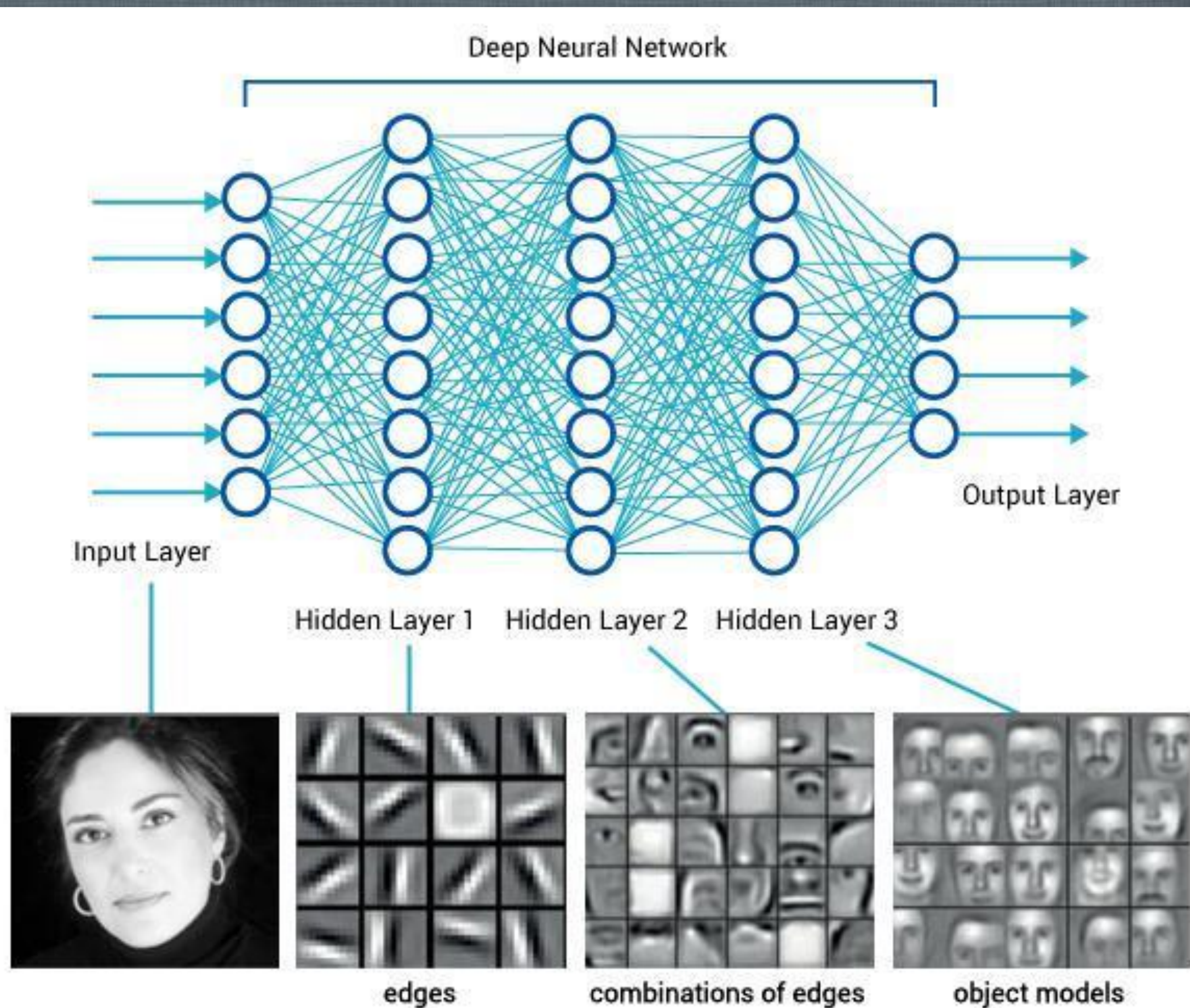
類神經網路

14

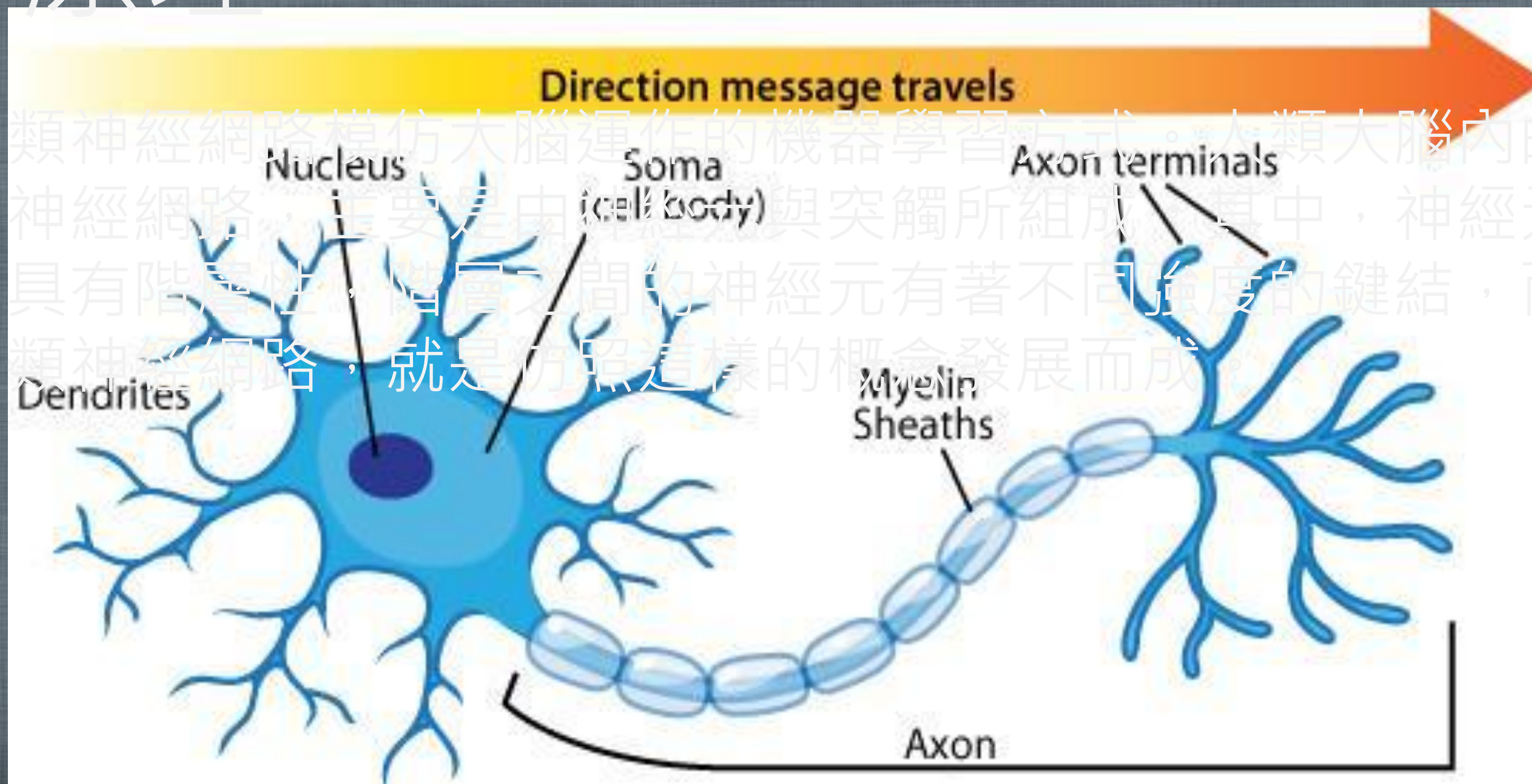
類神經網路，在機器學習和認知科學領域，是一種模仿生物神經網路（動物的中樞神經系統，特別是大腦）的結構和功能的數學模型或計算模型，用於對函式進行估計或近似。神經網路由大量的人工神經元聯結進行計算。大多數情況下人工神經網路能在外界資訊的基礎上改變內部結構，是一種自適應系統

簡介

能透過層層處理將大量無序的訊號漸漸轉為有用的資訊並解決問題。



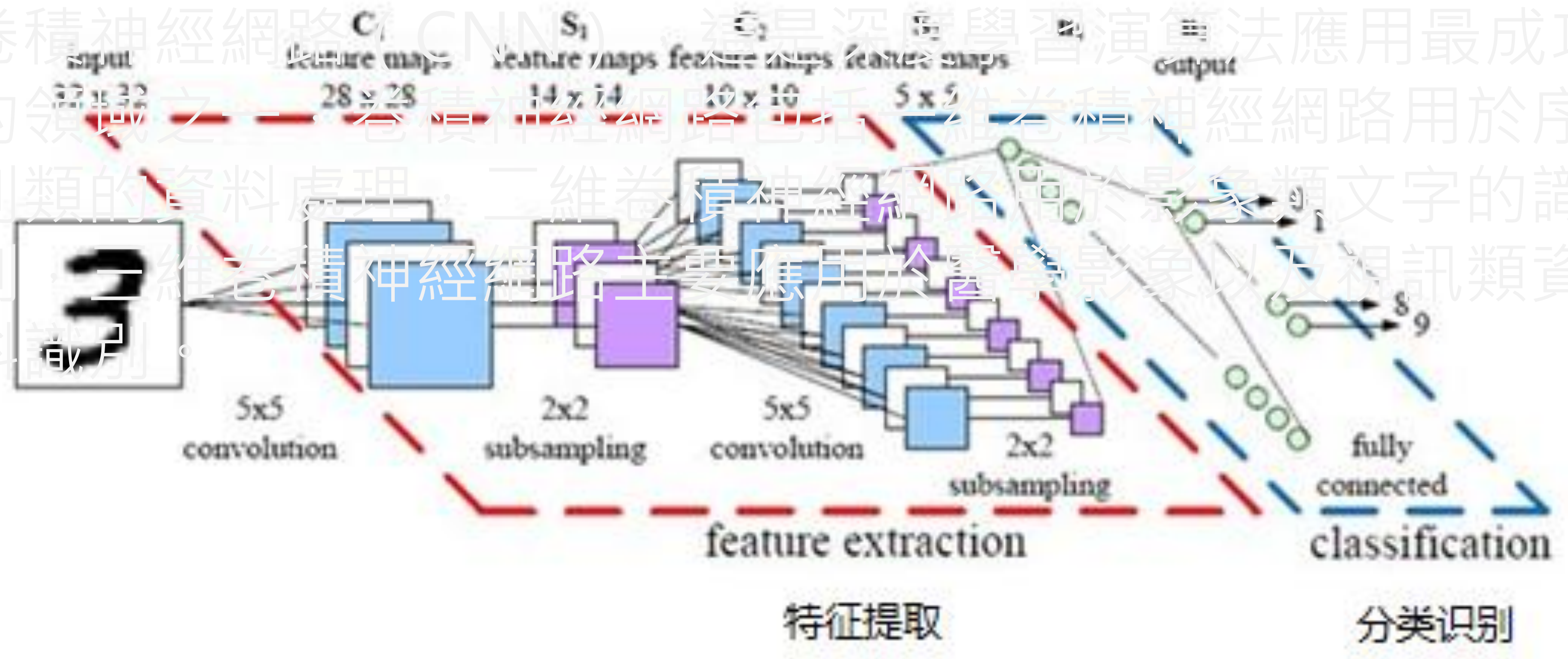
原理



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，就是仿照這樣的觀念發展而成。

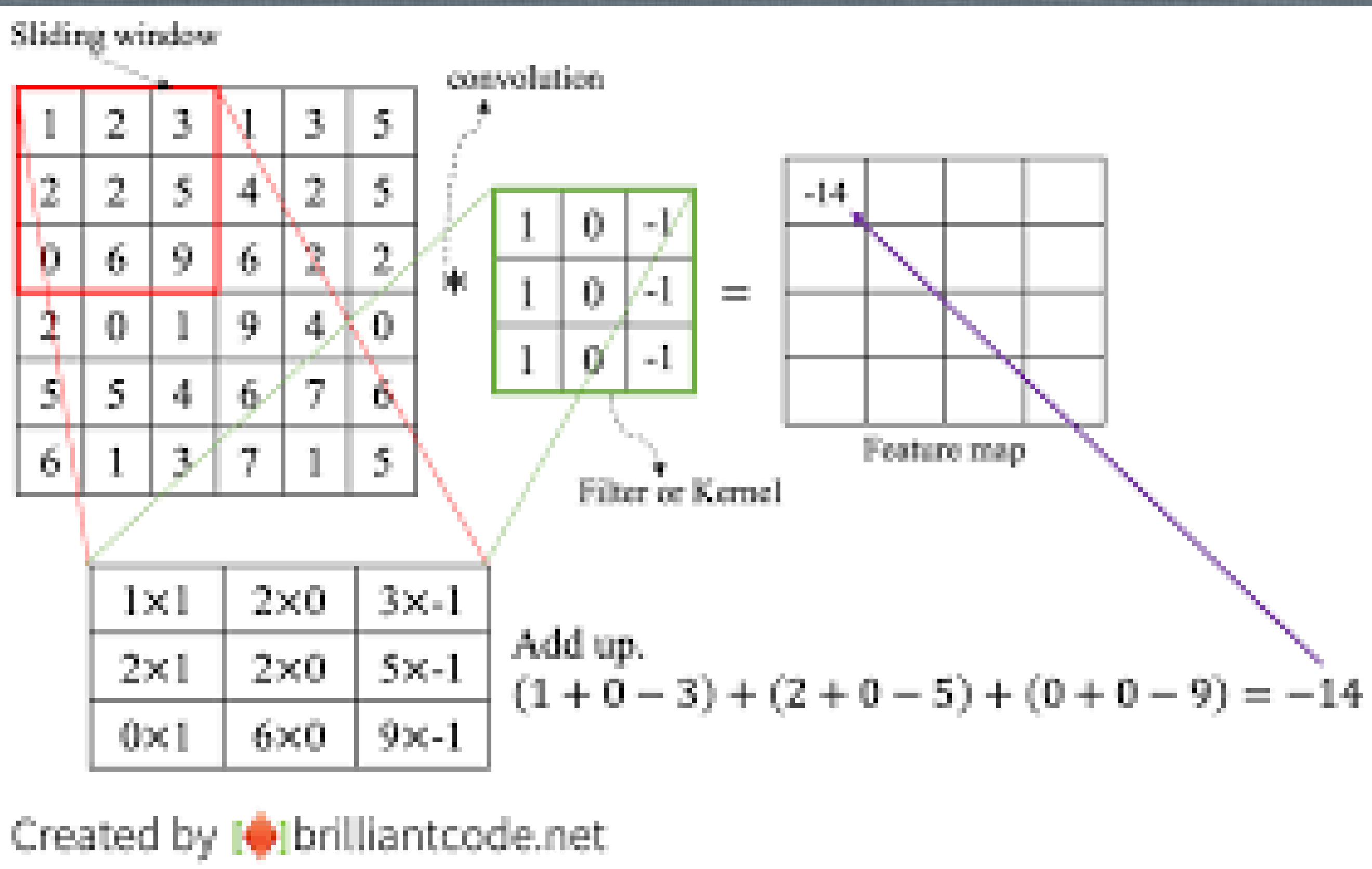
卷積神經網路

卷積神經網路 (CNN) 這是深度學習演算法應用最成功的領域之一。卷積神經網路包括一維卷積神經網路用於序列類的資料處理，二維卷積神經網路用於影像類文字的識別，三維卷積神經網路主要應用於醫學影像以及視訊類資料識別。



卷積層(Convolution Layer)

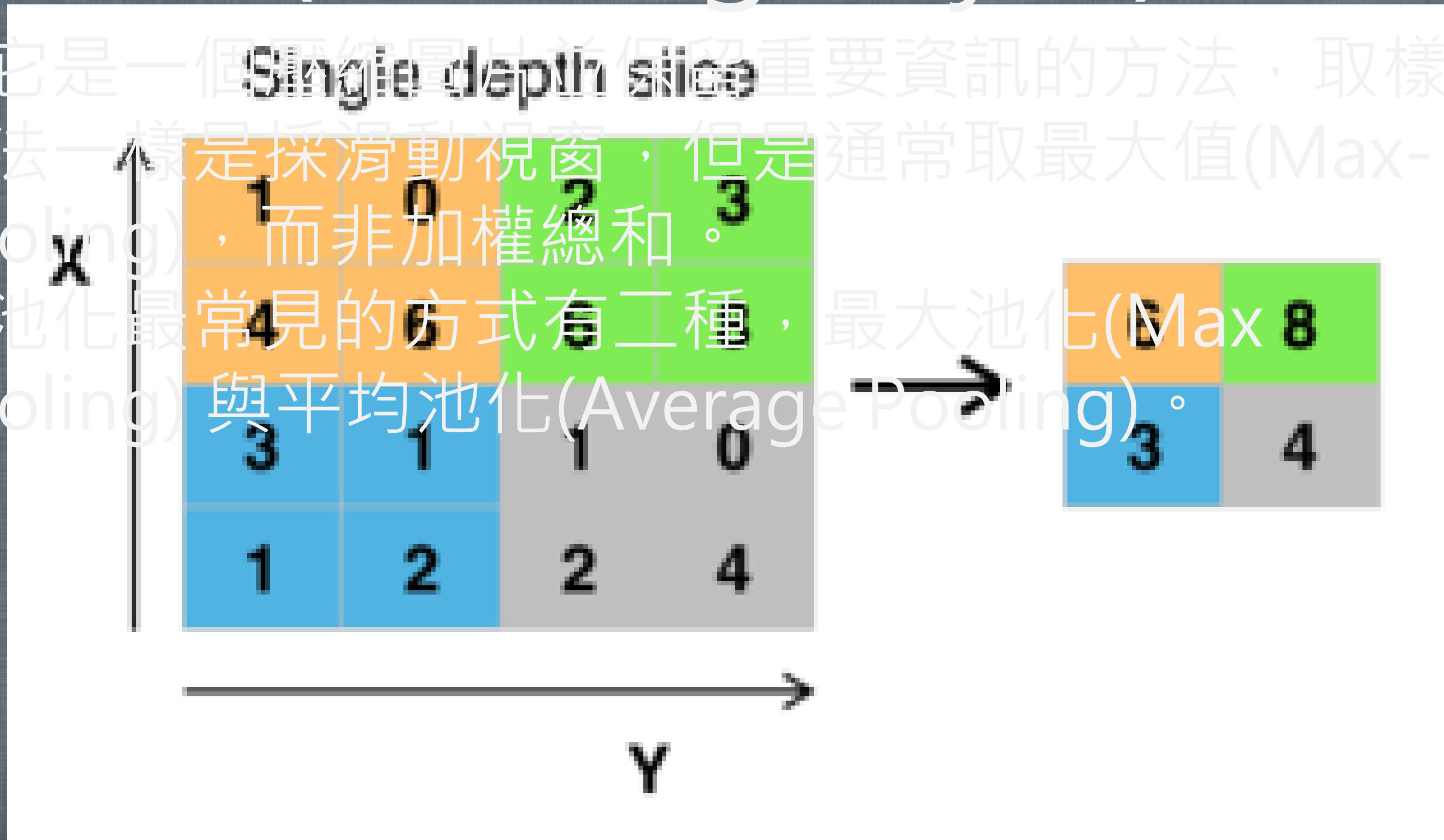
就塊塊就可



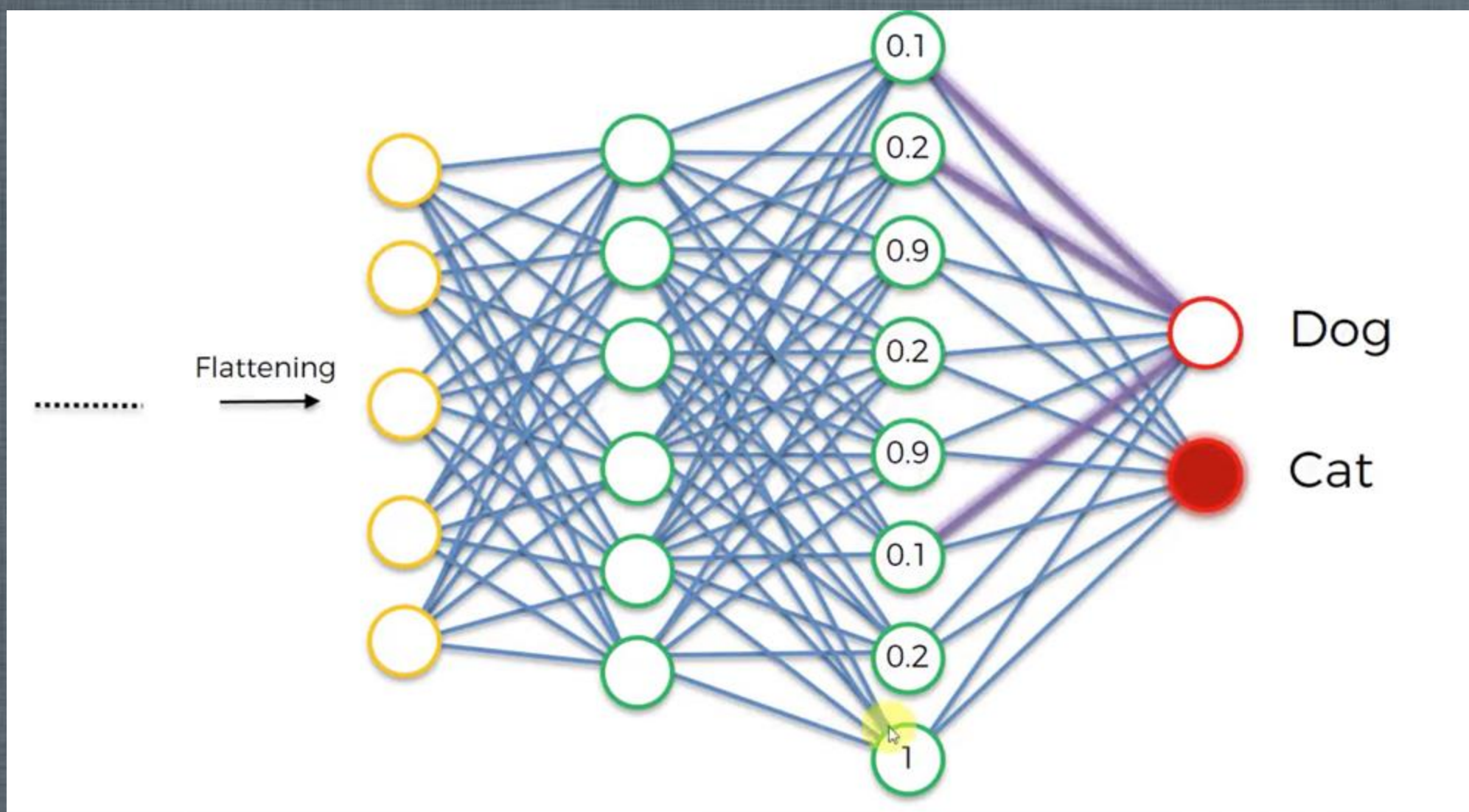
池化層(Pooling Layer)

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。

池化最常見的方式有二種，最大池化(Max Pooling)與平均池化(Average Pooling)。



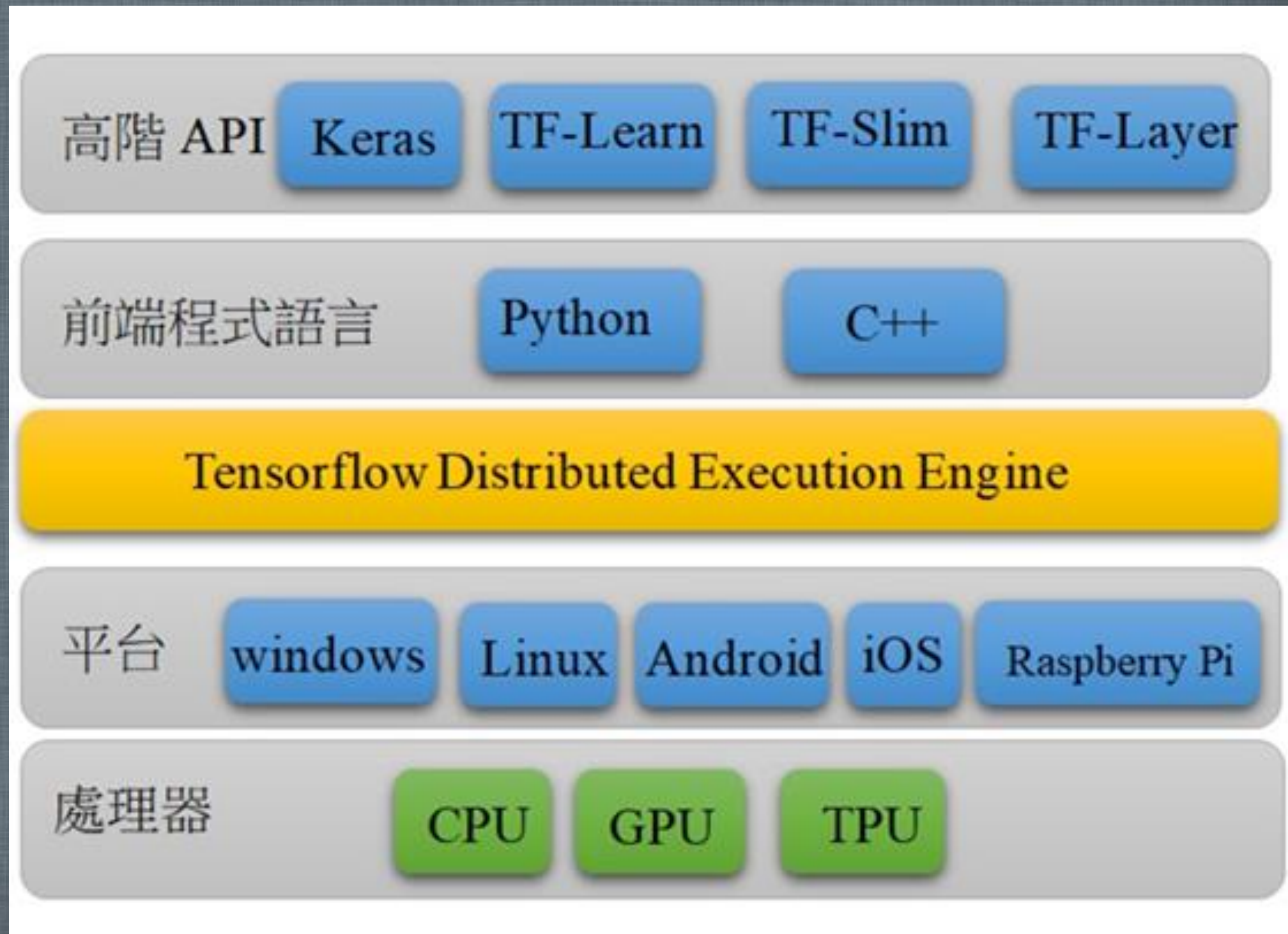
連接層



全連接層在整個卷積神經網路中起到“分類器”的作用，即通過卷積、啟用函式、池化等深度網路後，再經過全連接層對結果進行識別分類。

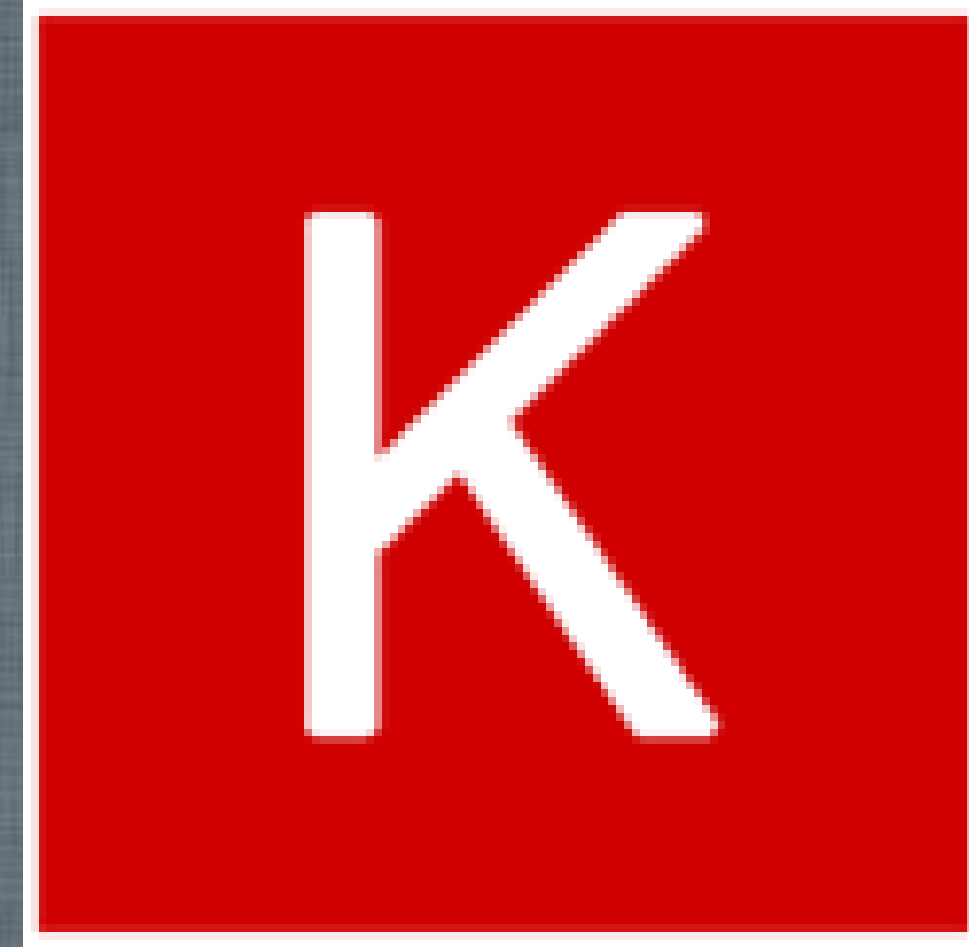
開源軟體庫

- tensorflow
- keras



keras

Keras是一個開放原始碼，基於Python高階深度學習的程式庫，它已經將訓練模型的輸入層、隱藏層、輸出層做好架構。





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
```


建立cnn各層

```
100 model = Sequential()#建立keras的Sequential模型
101 #建立卷積層1
102 model.add(Conv2D(filters=32,kernel_size=(3,3),input_shape=(128,128,3),activation='relu',padding='same'))
103 #加入Drop層1，避免overfitting
104 model.add(Dropout(rate=0.25))
9 model.compile(
0                                     metrics=['accuracy'])
1 #以样本数量生成model.save("Study_model.h5")
2 permutation = model.save_weights("Study_weight.h5") e[0]))
3 #按照随机生成的
4 x_img_train_nc #model.Load_weights("StudyTrain.h5")
5 #训练集有4个维
6 y_label_train # del model
116 model.add(Dropout(rate=0.25))
117 #建立隱藏層
118 model.add(Dense(1024,activation='relu'))
119 model.add(Dropout(rate=0.25))
120 #建立輸出層
121 model.add(Dense(3,activation='softmax'))
122 print(model.summary())
123
```


tensorflow

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。其中TFRecord是一種專為Tensorflow打造的儲存文件的格式。



初版程式(失敗)

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



自製資料庫(使用tensorflow)



準備圖片



製作tfrecords檔案



卷積神經網路訓練



何謂tfrecord?

一種將影像資料和標籤放在一起的二進位制檔案，能更好的利用記憶體，在tensorflow中快速的複製，移動，讀取，儲存等等..

tfrecord會根據你選擇輸入檔案的類，自動給每一類打上同樣的標籤

特點

1. 是Tensorflow官方通用的輸入格式

2. 能夠使用Tensorflow各項數據輸入API (包含圖片預處理、多執行序圖片輸入pipeline)，讓數據輸入的效率提升，訓練速度更快。



Python 讀取影像檔的方法

1. `PIL.Image.open`
2. `scipy.ndimage.imread`
3. `scipy.misc.imread`
4. `skimage.io.imread`
5. `cv2.imread`
6. `matplotlib.image.imread`


```
18 writer = tf.python_io.TFRecordWriter('studng.tfrecords') #要生成的檔案
19 #对应的列表中，同时贴上标签，存放到Label列表中。
20 for
21 def get_files(file_dir, ratio):
22     for file in os.listdir(file_dir+'/sleep'):
23         studying.append(file_dir + '/sleep'+ '/' + file)
24         label_sleep.append(1)
25     for file in os.listdir(file_dir+'/playing'):
26         sleep.append(file_dir + '/playing'+ '/' +file)
27         label_playing.append(0)
28     for file in os.listdir(file_dir+'/studying'):
29         playing.append(file_dir + '/studying'+ '/' + file) ),
30         label_studying.append(2) 1))
31
32
33
34
35
36
37 plt.show()
```


各檔案格式比較

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

比較

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那麼快。

Tensorflow(tf)

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

Section 4

遭遇困難


```
Thonny - /home/pi/Study/Study_Pred.py @ 64 1
w Load Save Run Debug Over Into Out Stop Zoom Quit
Pred.py x
5 img_normalize=img.astype('float32')/255.0
6 img_normalize= img_normalize[np.newaxis,...]
7
8 x_img_test_normalize=img_normalize.astype('float32')/255.0
9 y_pred1 = model_try1.predict(x_img_test_normalize)
10 y_pred2 = model_try2.predict(x_img_test_normalize)
11
12 if np.argmax(y_pred1[0]) == 0:
13
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/__init__.py:98: The name tf.COMPIILER_VERSION is deprecated. Please use tf.version.COMPIILER_VERSION instead.
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/__init__.py:98: The name tf.CXX11_ABI_FLAG is deprecated. Please use tf.sysconfig.CXX11_ABI_FLAG instead.
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/__init__.py:98: The name tf.ConditionalAccumulator is deprecated. Please use tf.compat.v1.ConditionalAccumulator instead.
Using TensorFlow backend.
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/keras/backend/tensorflow_backend.py:4070: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.
2019-12-24 15:18:00.601347: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 268435456 exceeds 10% of system memory.
Backend terminated or disconnected. Use 'Stop/Restart' to restart ...
Study Thonny - /home/pi/ python-Tensorflow分... 1524
```


Section 5

成果

Thingspeak

ThingSpeak是一種開源的物聯網（IoT）應用程序和API，可通過Internet或通過局域網使用HTTP和MQTT協議來存儲和檢索來自物聯網的數據。

The logo consists of a white speech bubble icon with a tail pointing towards the bottom-left, followed by the text "ThingSpeak" in a white, bold, sans-serif font. A thin white horizontal line is positioned directly beneath the text.

ThingSpeak

程式碼

37

```
1 import random
2 import requests
3 import time
4
5 for count in range(3):
6     print('start of the request')
7     myurl = 'https://api.thingspeak.com/update?api_key=S2G4I7CNSVJF7AYK'
8     myurl += str((random.randint(1,15)))
9     request=requests.get(myurl)
10    print('request sent. yougot this answer:')
11    print(request.text)
12    time.sleep(15)
13
```


學習成效

Channel ID: 949435

Author: mwa0000017056960

Access: Private

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

[Add Visualizations](#)

[Add Widgets](#)

[Export recent data](#)

[MATLAB Analysis](#)

[MATLAB Visualization](#)

Channel 4 of 4 < >

Channel Stats

Created: [less than a minute ago](#)

Entries: 0



This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our [Privacy Policy](#) to learn more about cookies and how to change your settings.



Section 6

未來期許

That's all!

Thank you! 😊
Any questions?