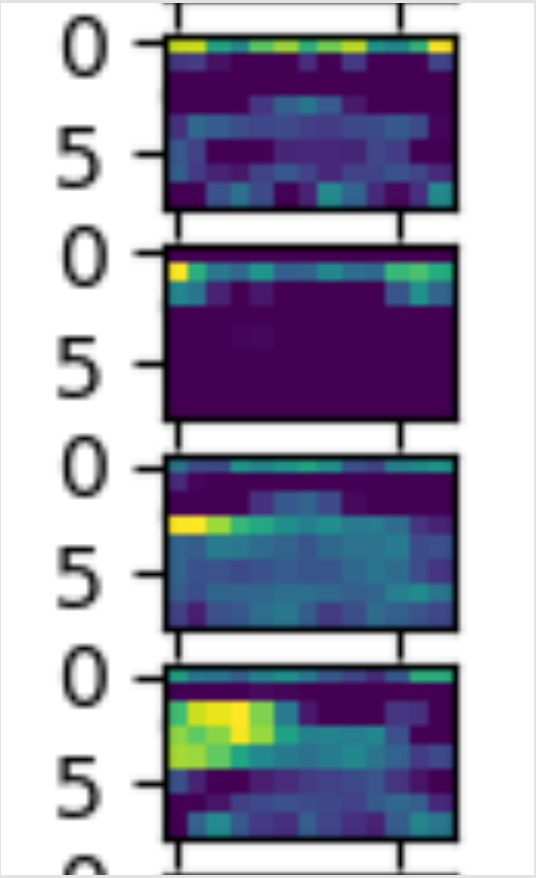
“和"我一起玩AI

-影像辨識”駕”厲害

臺南市和順國中-林信廷



講師介紹

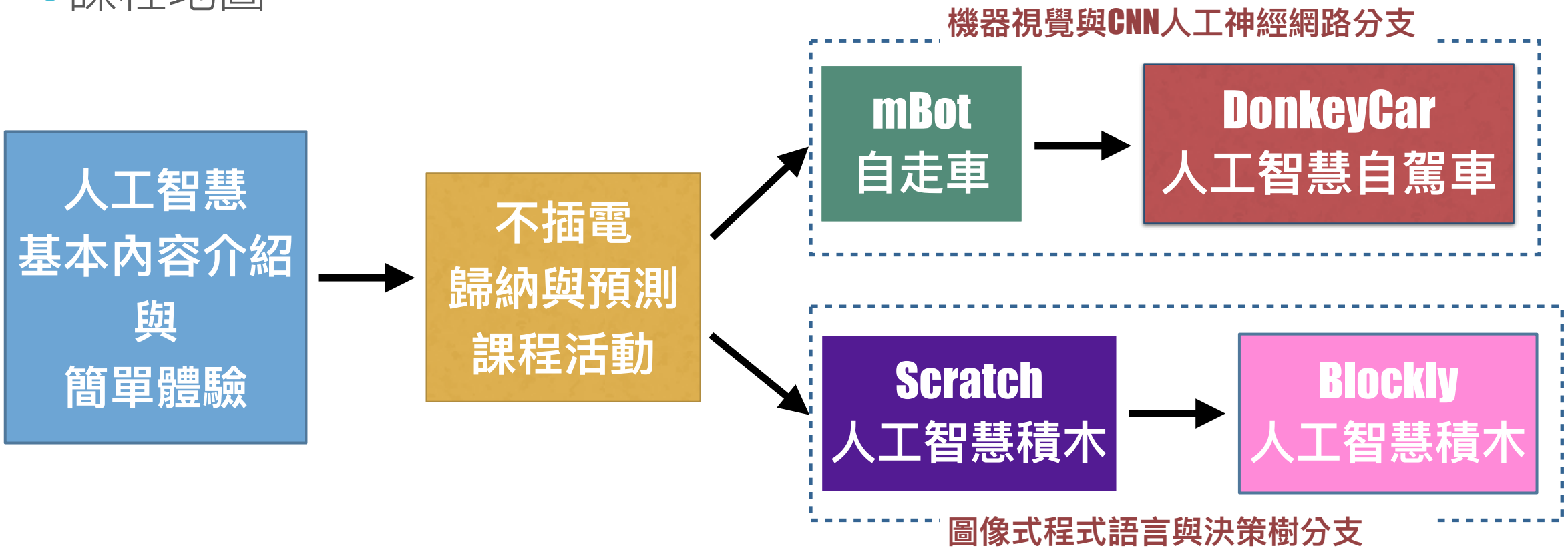
- 臺南市和順國中資訊組長
- 臺南市資訊組長分區召集人
- 臺南市科技輔導團兼任輔導員
- 和順科技中心課程設計教師

- 專長：機電整合、人工智慧



課程地圖

- 課程地圖：


人工智慧
基本內容介紹
與
簡單體驗
不插電
歸納與預測
課程活動
機器視覺與CNN人工神經網路分支
mBot
自走車
DonkeyCar
人工智慧自駕車
Scratch
人工智慧積木
Blockly
人工智慧積木
圖像式程式語言與決策樹分支

- 簡單的體驗
  - https://teachablemachine.withgoogle.com/train：影像、聲音、肢體辨識
  - https://ml-playground.com：座標點分類器
  - https://www.cs.ryerson.ca/~aharley/vis/conv/：手寫數字辨識與CNN
  - https://codinglab.tw/aiplayground：類似teachable machine
  - https://courses.codinglab.tw/p/ai-platform：類似teachable machin

- 人工智慧基本內容介紹：和AI做朋友，by教育部
  - 紙本書籍，分為國小、國中、高中用，有很多教學活動
  - 中華開放教育平台：https://www.openedu.tw/course.jsp?id=876



和AI做朋友 - 相逢篇：人工智慧有意思 (教材)



和AI做朋友 - 相逢篇：人工智慧有意思 (教案)



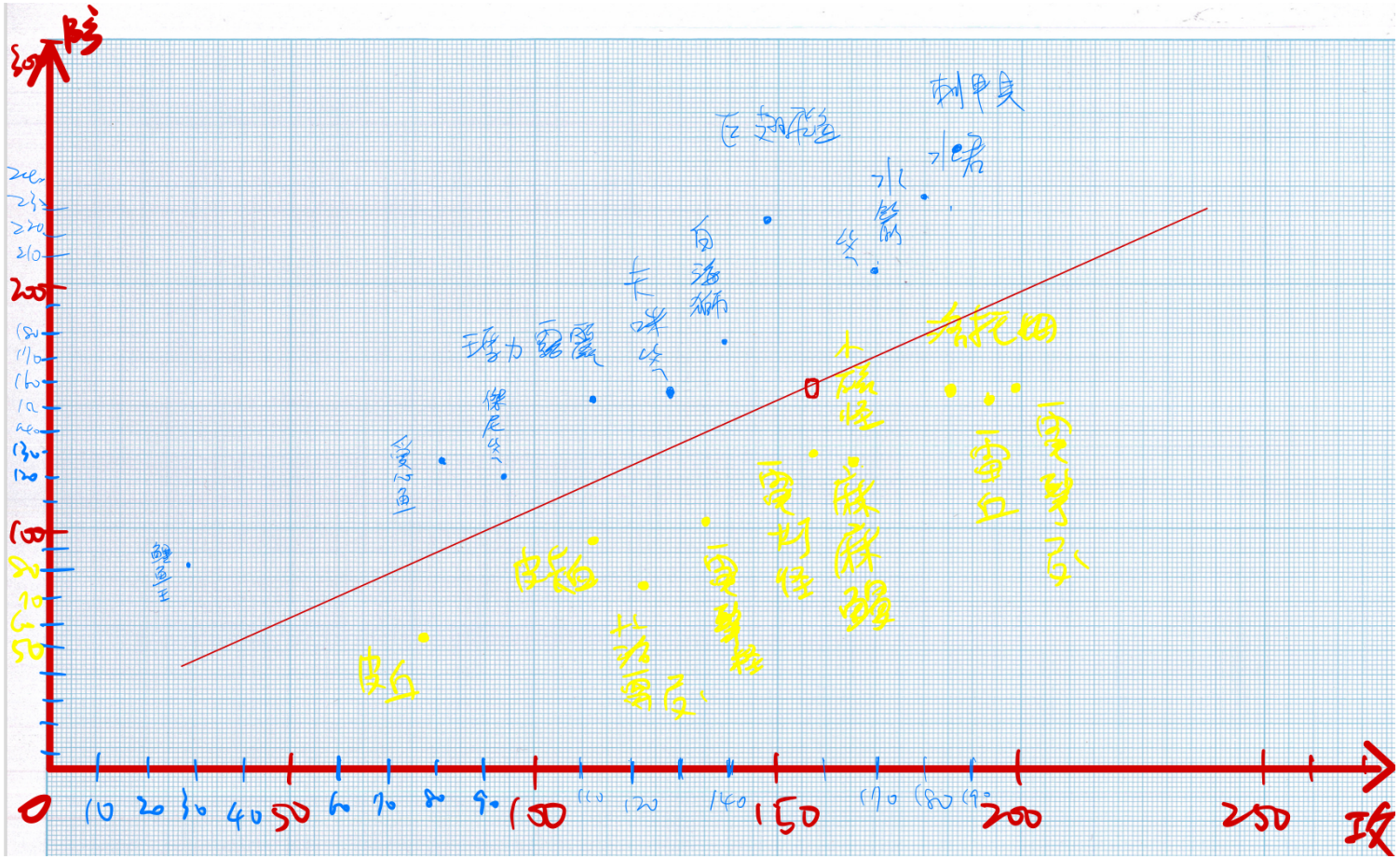
和AI做朋友 - 相識篇：開啟AI任意門 (教材)

- 簡單的體驗
  - https://teachablemachine.withgoogle.com/train：影像、聲音、肢體辨識
  - https://ml-playground.com：座標點分類器
  - https://www.cs.ryerson.ca/~aharley/vis/conv/：手寫數字辨識與CNN
  - https://codinglab.tw/aiplayground：類似teachable machine
  - https://courses.codinglab.tw/p/ai-platform：類似teachable machin

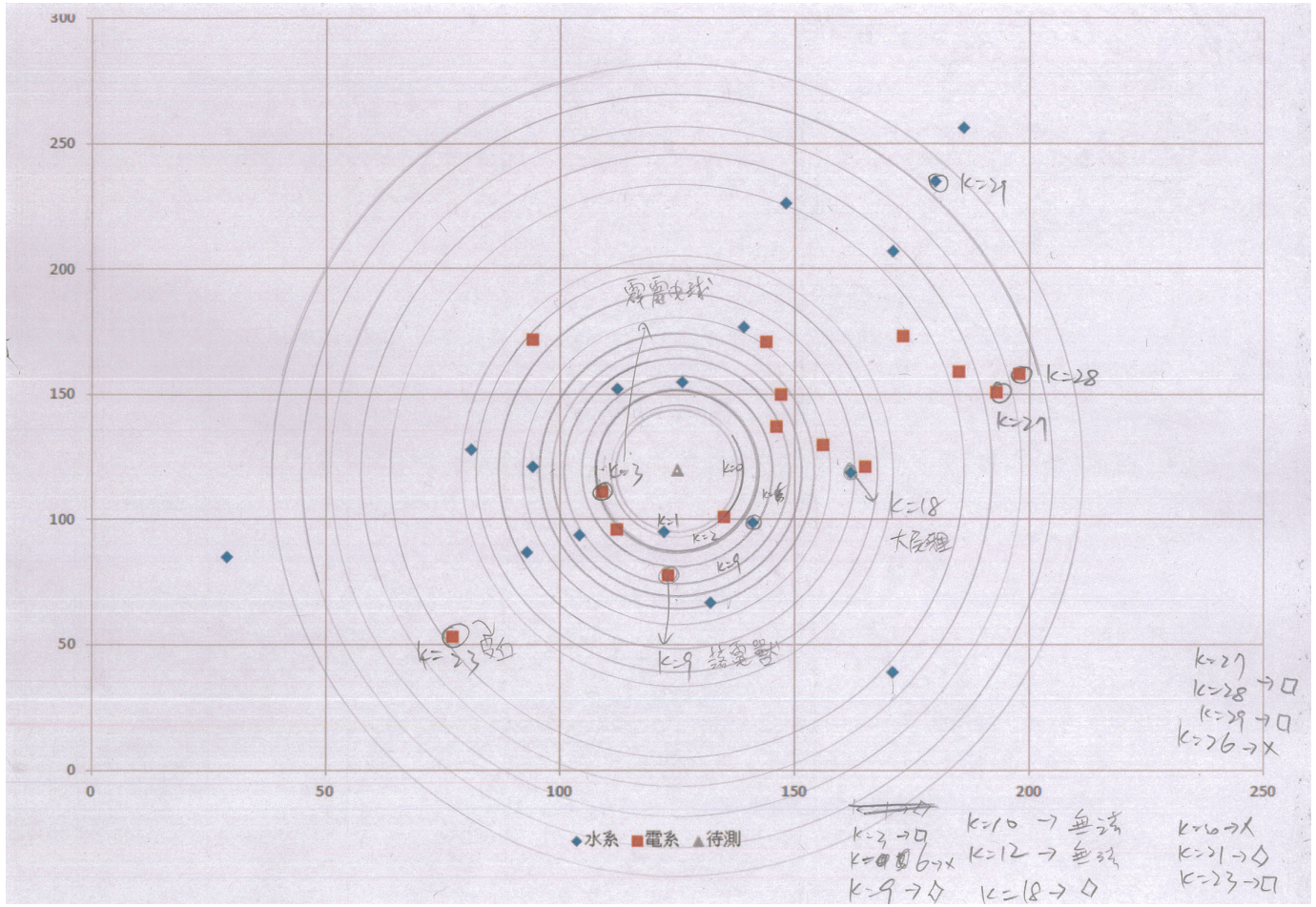
- 不插電的歸納與預測課程活動：
  - 和中教師校園大亂鬥
  - 教師的卡牌上有攻擊力與防禦力，學生用這些數值分類老師，並預測新的卡牌的能力



- 不插電的歸納與預測課程活動：

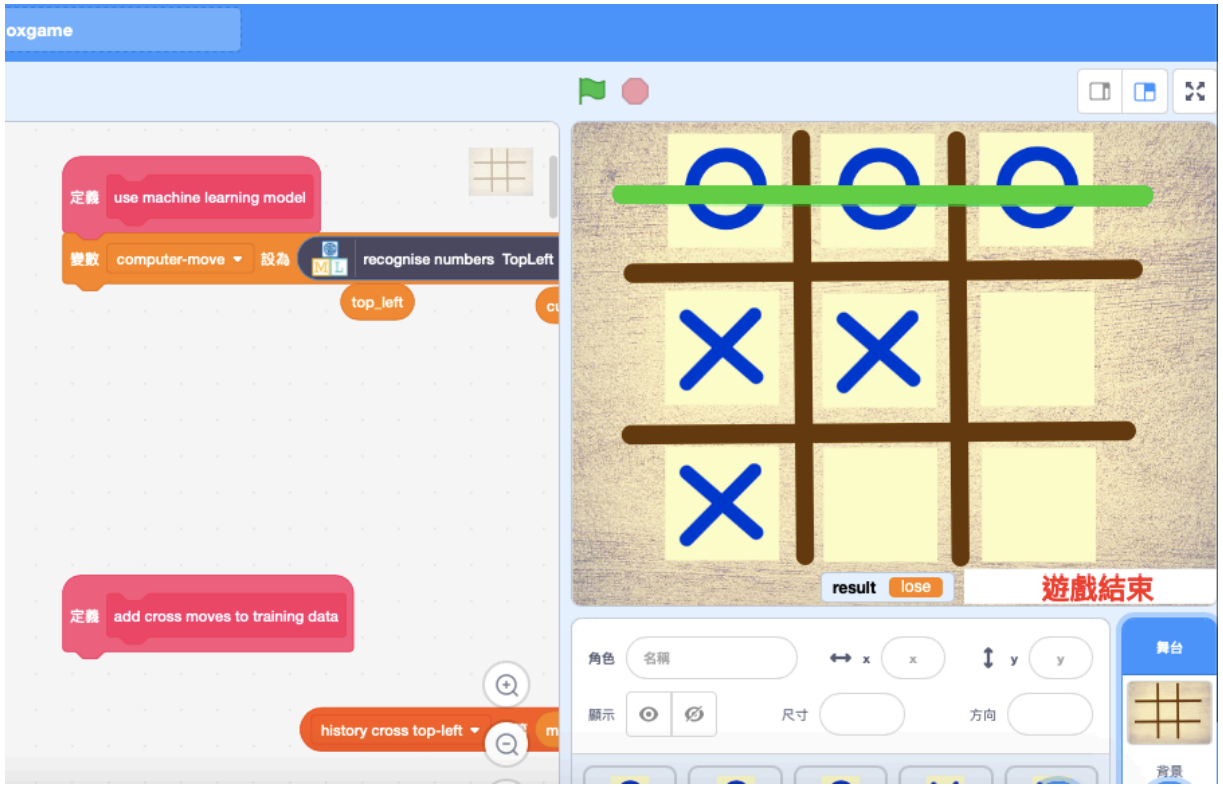


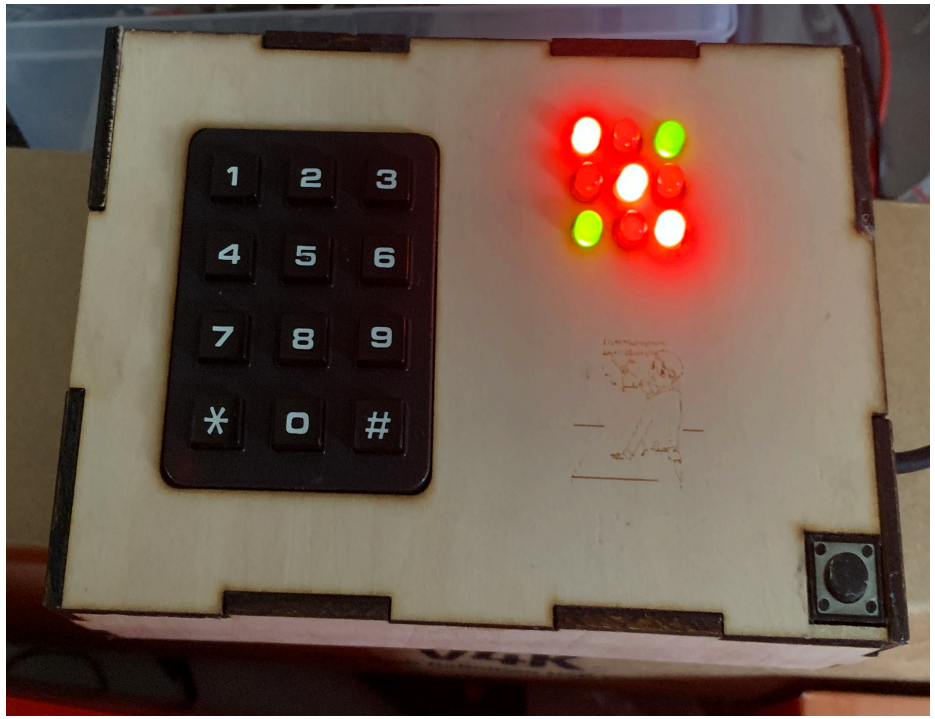
- 不插電的歸納與預測課程活動：



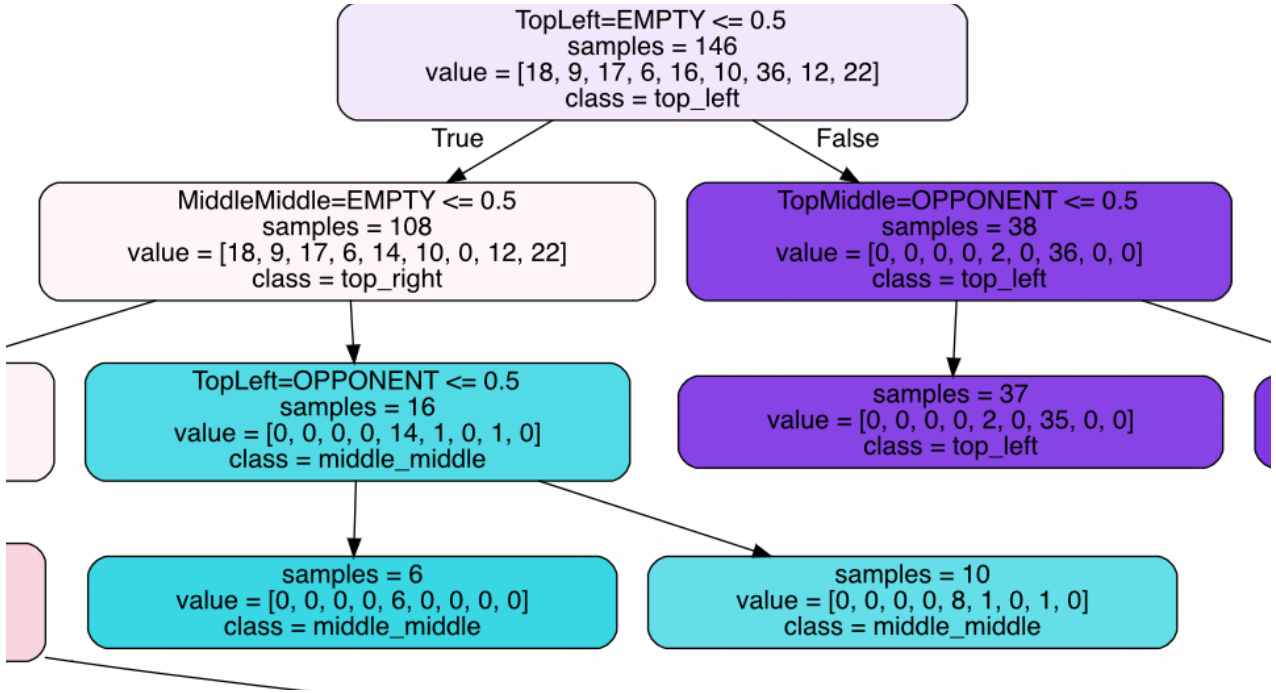
- SCRATCH分支：
  - 國中生皆會受到圖像式程式語言的訓練，可以跟課程結合
  - 自由度很高，自己寫的程式，所以可以自己決定要讓機器學習的資料
  - 學習決策樹的相關概念
  - 以「AlphaGO」作為引起興趣的元素
  - Blockly有更高階的應用，但用法不太一樣

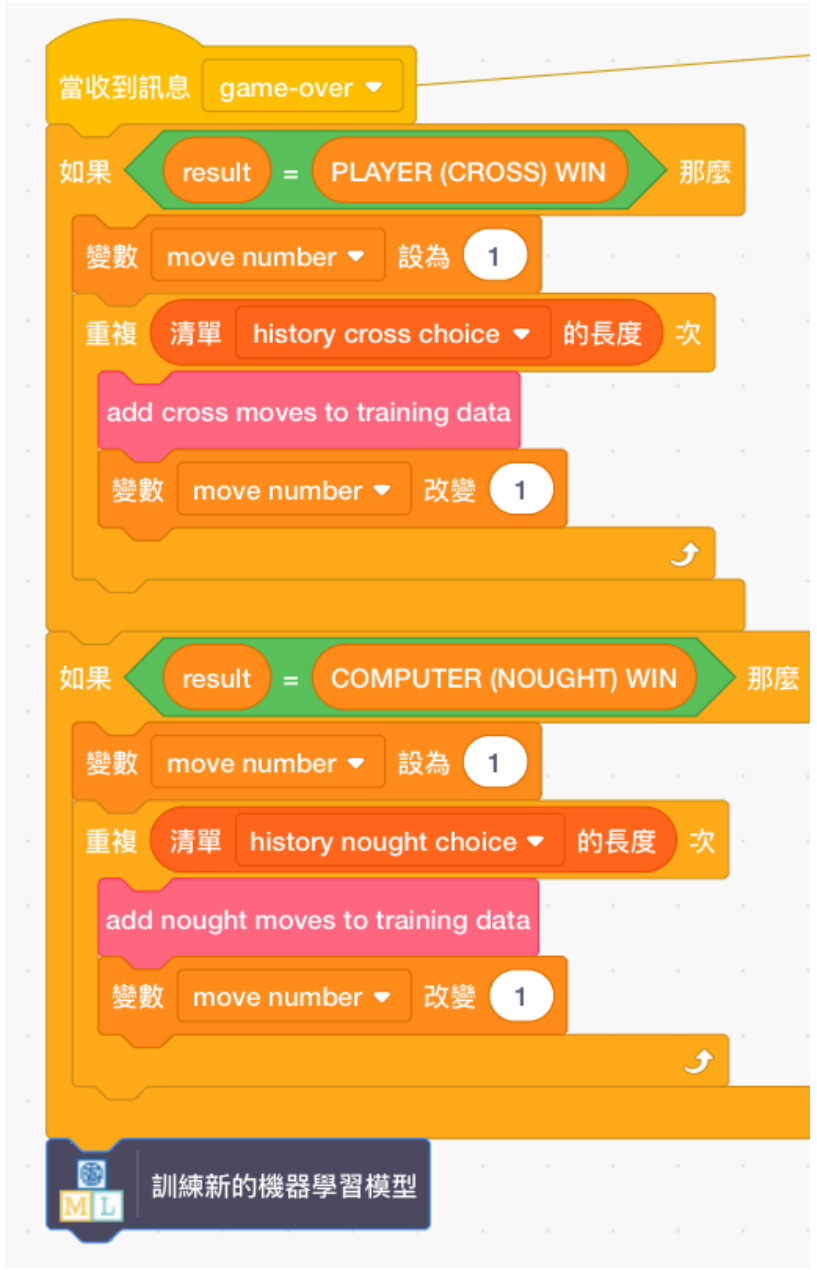
- Scrtach分支：人類規則智慧與人工智慧的比較





- Scrtach分支


TopLeft=EMPTY <= 0.5
samples = 146
value = [18, 9, 17, 6, 16, 10, 36, 12, 22]
class = top_left
True
False
MiddleMiddle=EMPTY <= 0.5
samples = 108
value = [18, 9, 17, 6, 14, 10, 0, 12, 22]
class = top_right
TopMiddle=OPPONENT <= 0.5
samples = 38
value = [0, 0, 0, 0, 2, 0, 36, 0, 0]
class = top_left
TopLeft=OPPONENT <= 0.5
samples = 16
value = [0, 0, 0, 0, 14, 1, 0, 1, 0]
class = middle_middle
samples = 37
value = [0, 0, 0, 0, 2, 0, 35, 0, 0]
class = top_left
samples = 6
value = [0, 0, 0, 0, 6, 0, 0, 0, 0]
class = middle_middle
samples = 10
value = [0, 0, 0, 0, 8, 1, 0, 1, 0]
class = middle_middle


當收到訊息 game-over
如果 result = PLAYER (CROSS) WIN 那麼
變數 move number 設為 1
重複 清單 history cross choice 的長度 次
add cross moves to training data
變數 move number 改變 1
如果 result = COMPUTER (NOUGHT) WIN 那麼
變數 move number 設為 1
重複 清單 history nought choice 的長度 次
add nought moves to training data
變數 move number 改變 1
訓練新的機器學習模型

- Donkeycar分支：
  - mBot的操作，讓學生體驗以人類規則控制的自駕車
  - Donkeycar的操作，讓學生體驗影像辨識的人工智慧自駕車

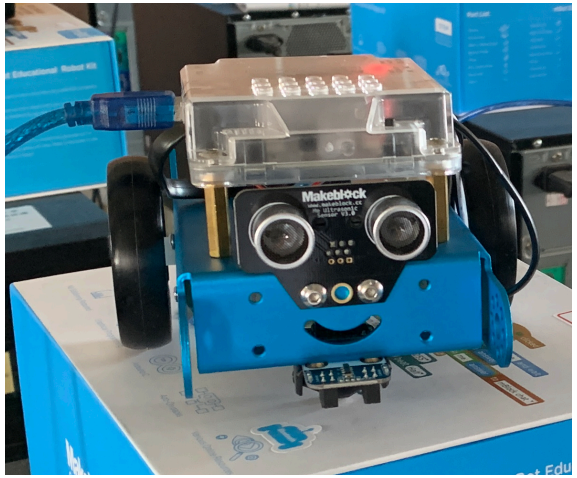
  } 比較

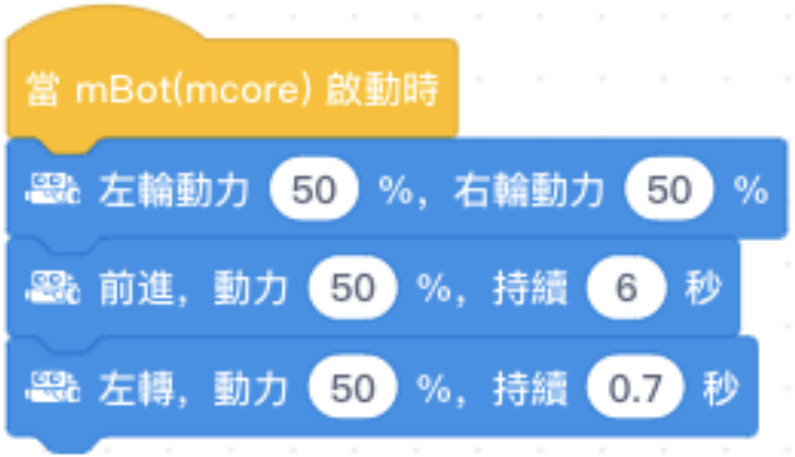
  - 了解機器的視覺，以及CNN卷積神經網路的相關知識
  - 以「特斯拉」作為引起興趣的元素

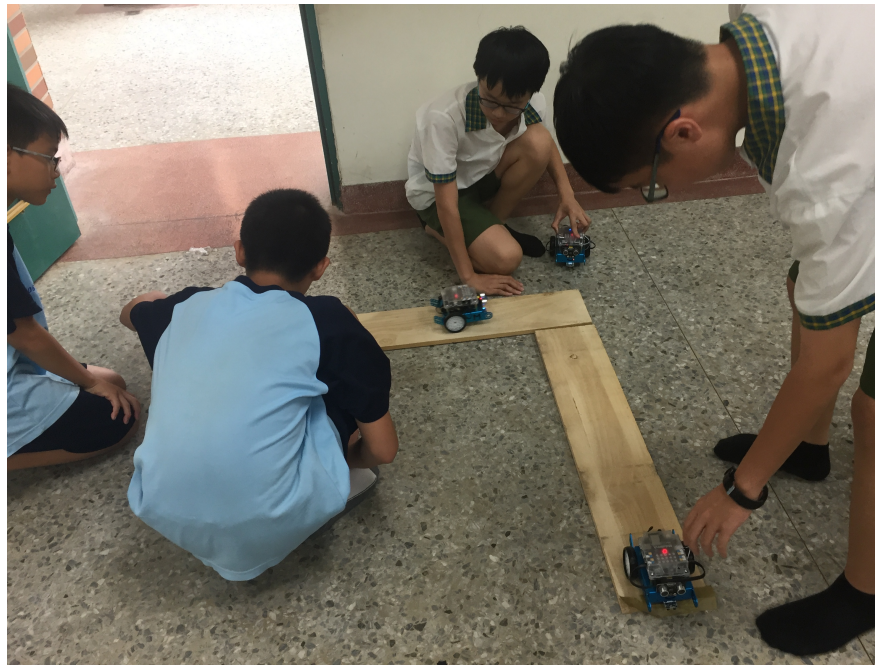
mBot小車

mBot小車

- 可以用外接的感測器
  - 超音波測距離、紅外線尋跡
- 也可以寫好輪子轉的時間

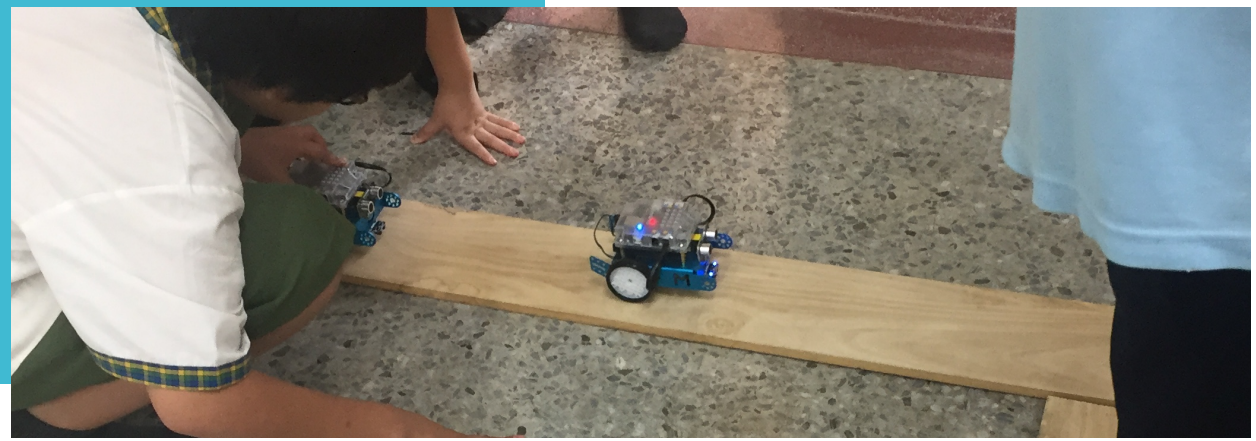


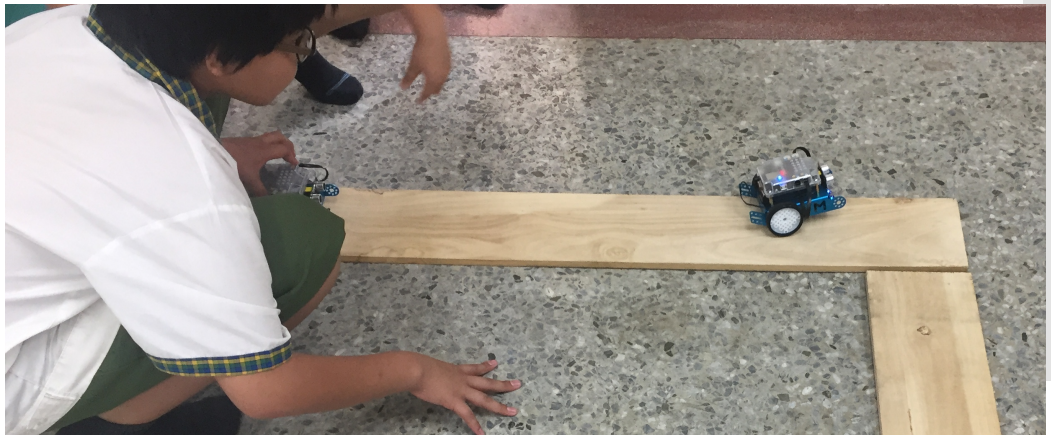




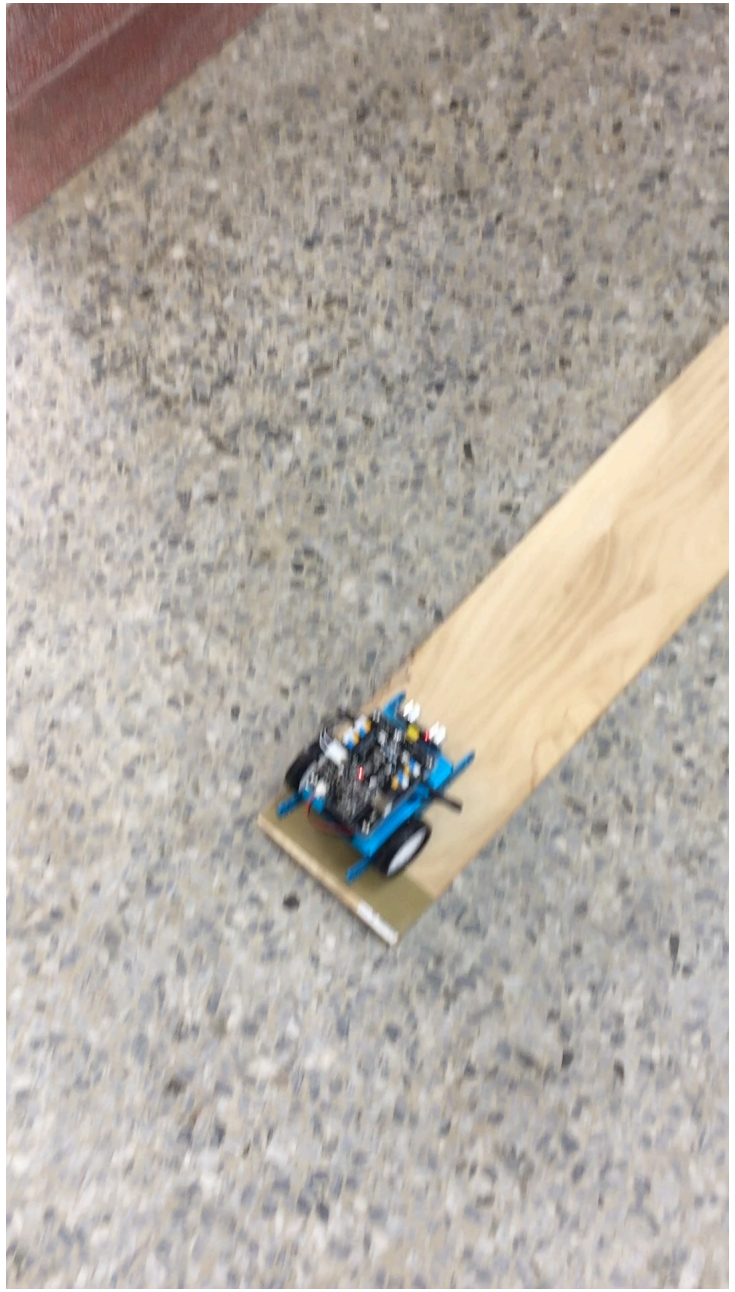
mBot小車

- 但真的有夠難控制
  - 兩邊輪子摩擦力不一樣
  - 電池電量會影響輸出能量
  - 齒輪組磨損造成誤差
  - 放置的位置與角度稍微不同，就整段壞光光





mBot小車



DonkeyCar影像辨識自駕車

DonkeyCar
影像辨識自駕車

- AI影像辨識車



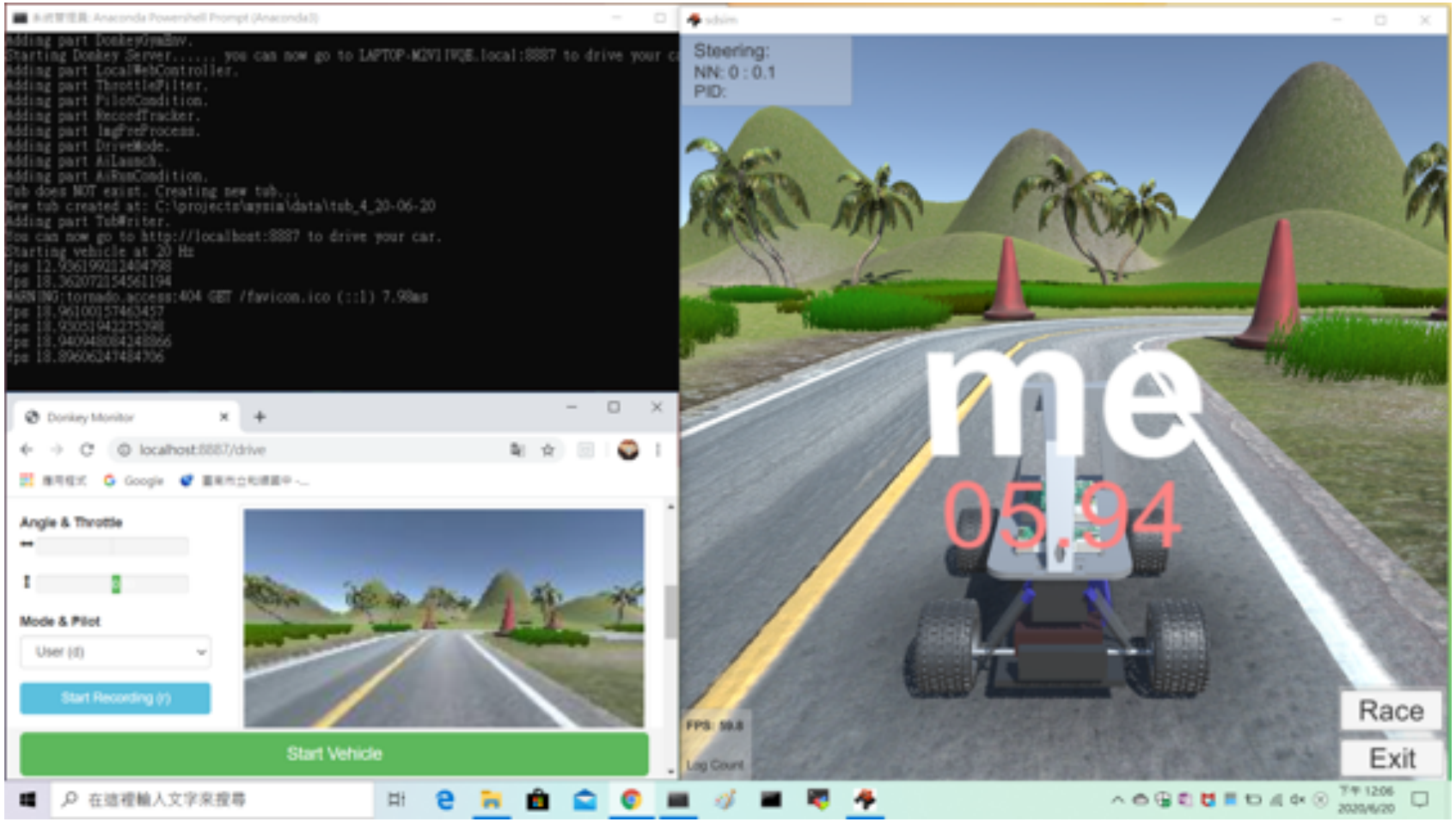
DonkeyCar
影像辨識自駕車

- AI影像辨識車



DonkeyCar
影像辨識自駕車

- AI影像辨識車：模擬器



DonkeyCar
影像辨識自駕車

- AI影像辨識車：模擬器可以用ps3搖桿控制

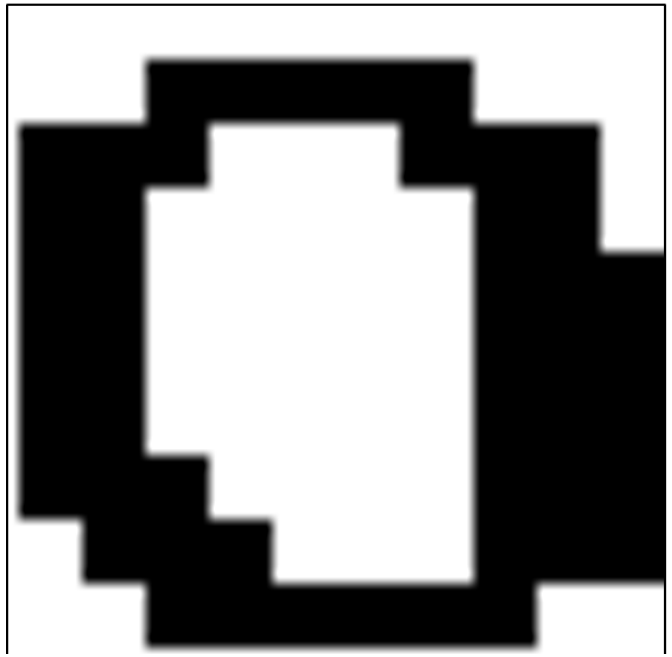


人工智慧與機器視覺

人工智慧
與
機器視覺

- 機器視覺：
  - 10*10的黑白圖片，是由100個像素構成的
  - 每個像素都有一個數值

原始影像



數位化的值

255	255	255	255	255	255	255	255	255	255
255	255	0	0	0	0	0	255	255	255
0	0	0	255	255	255	0	0	0	255
0	0	255	255	255	255	255	0	0	255
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0
255	0	0	0	255	255	255	0	0	0
255	255	0	0	0	0	0	0	255	255

人工智慧與機器視覺

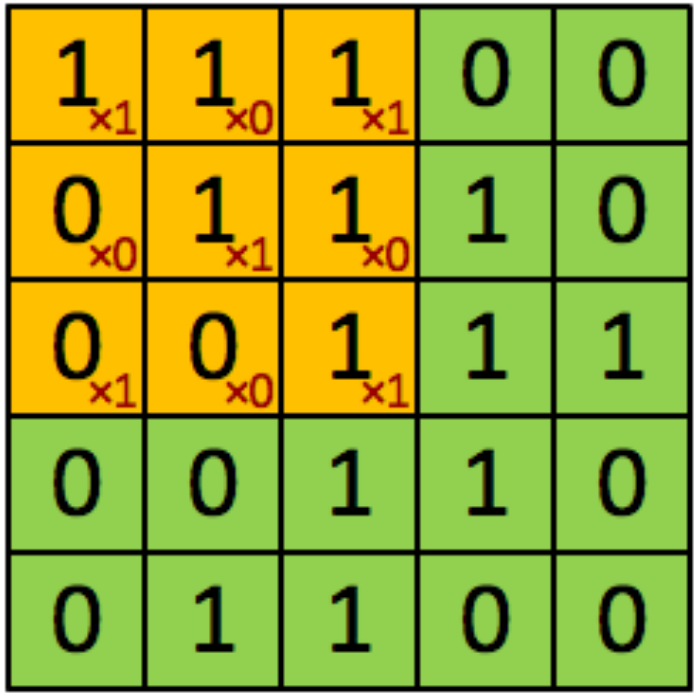
- 機器視覺：
  - 把所有像素的數值直接丟進去學習、分類，好嗎？
- 人工智慧是要找出特徵，每一個像素就是一個特徵嗎？

人工智慧與機器視覺

- CNN（Convolutional Neural Network）:卷積神經網路
  - 卷積的技術可以減輕類神經網路的負擔
  - 人類的視覺本身就比較像卷積的方式看東西

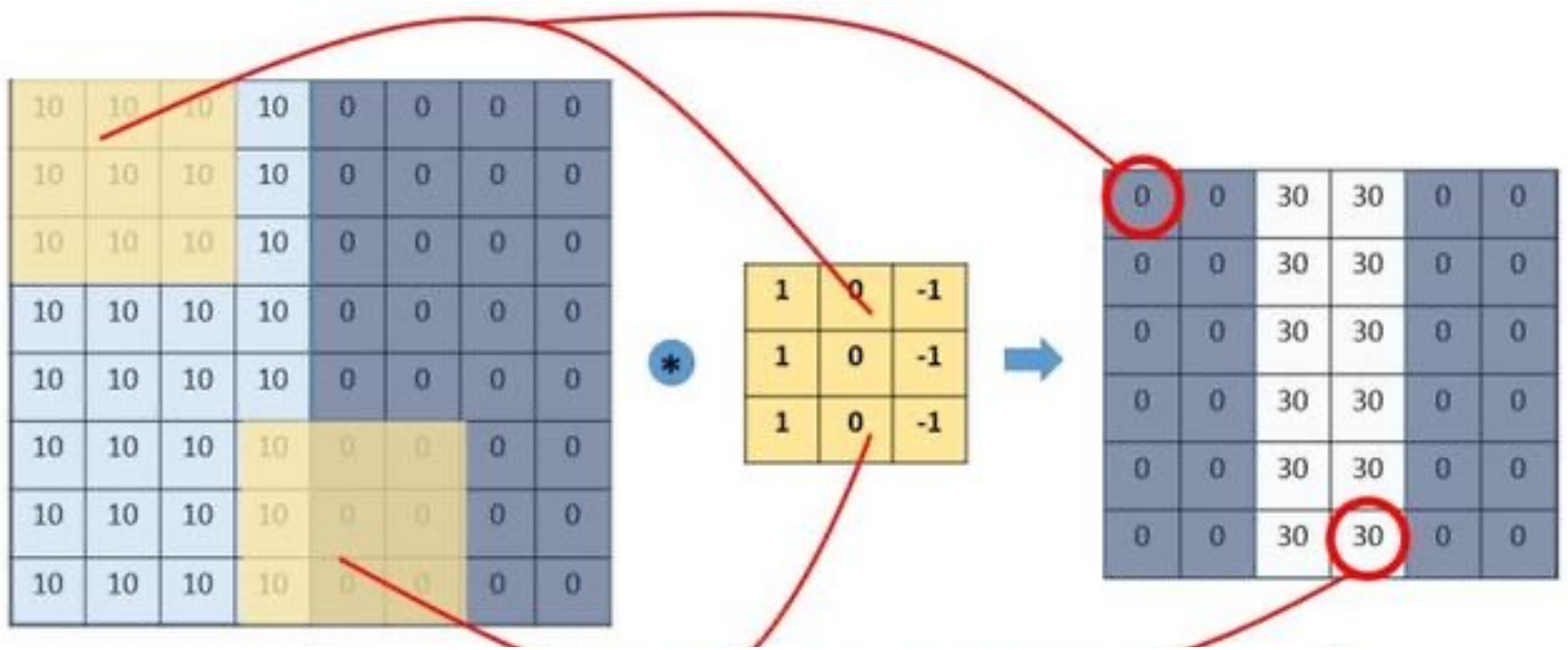
人工智慧
與
機器視覺

- 卷積：把圖片的一個區塊濃縮成一個值
  - 先經過convolution kernel計算矩陣
  - 再總加矩陣，作為特徵值，處理後的圖片稱為feature map



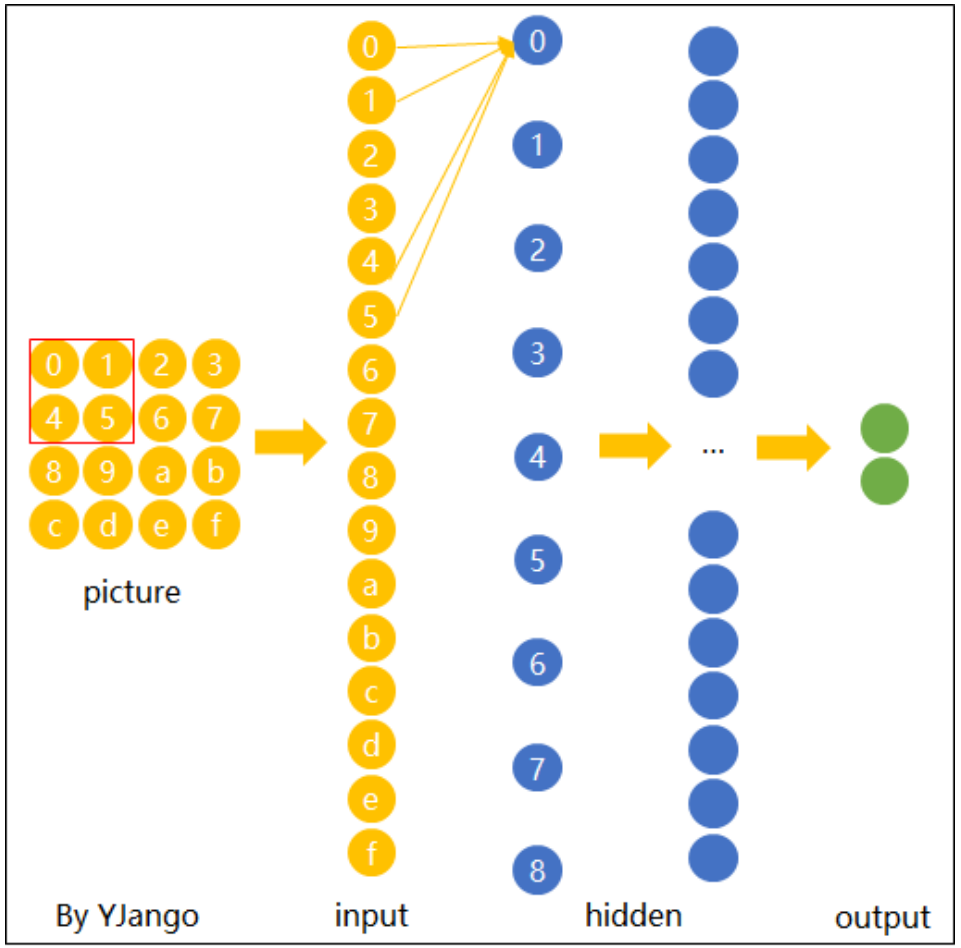
人工智慧
與
機器視覺

- CNN:卷積類神經網路
  - 好的kernel可以幫我們找出特徵



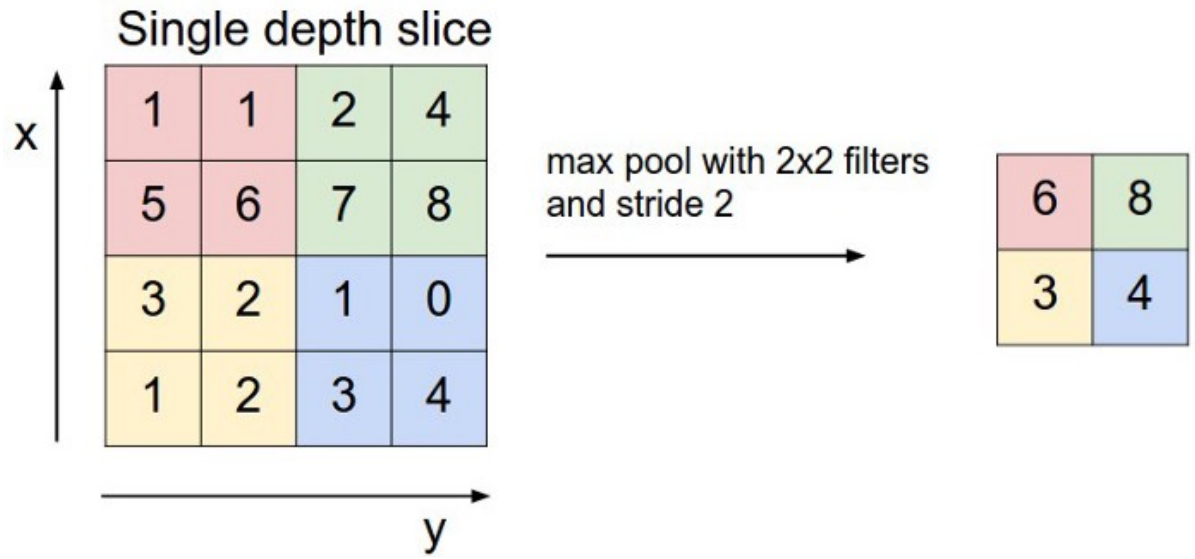
人工智慧
與
機器視覺

- CNN:卷積類神經網路
  - 卷積處理的神經連結
  - 部分連接層



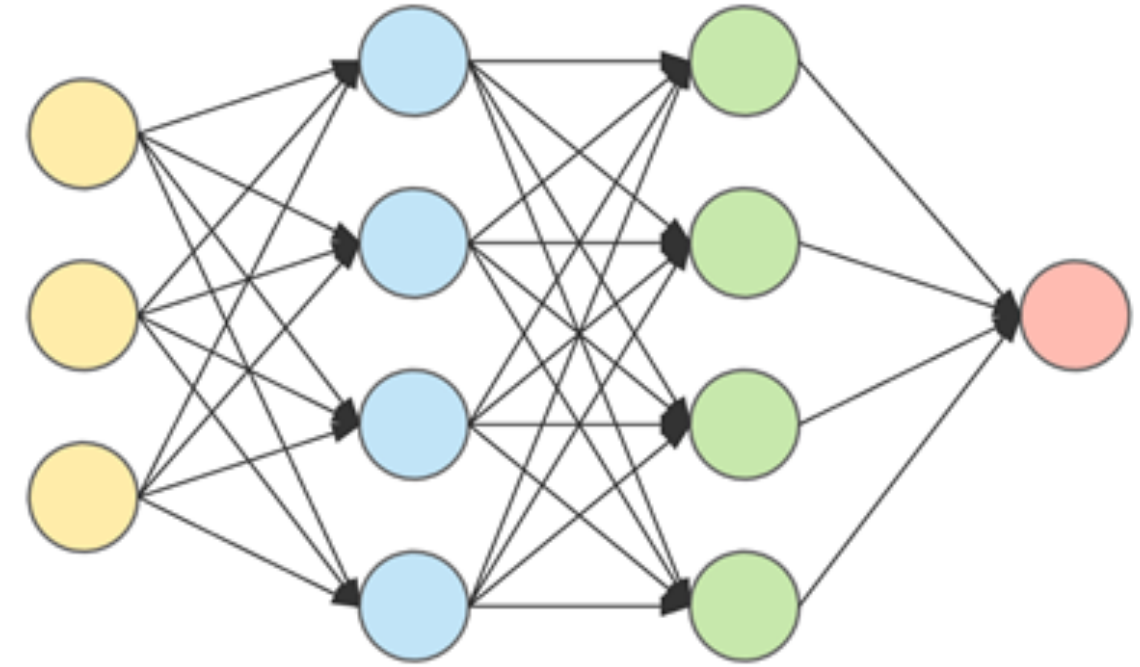
人工智慧與機器視覺

- CNN:卷積類神經網路
  - pooling（池化）：保留特徵
    - average pooling：定義一部分的矩陣，所有值相加，取平均，會把圖片再變小
    - max pooling：較常用，也有非常好的表現。因為失去的資訊往往不太重要。



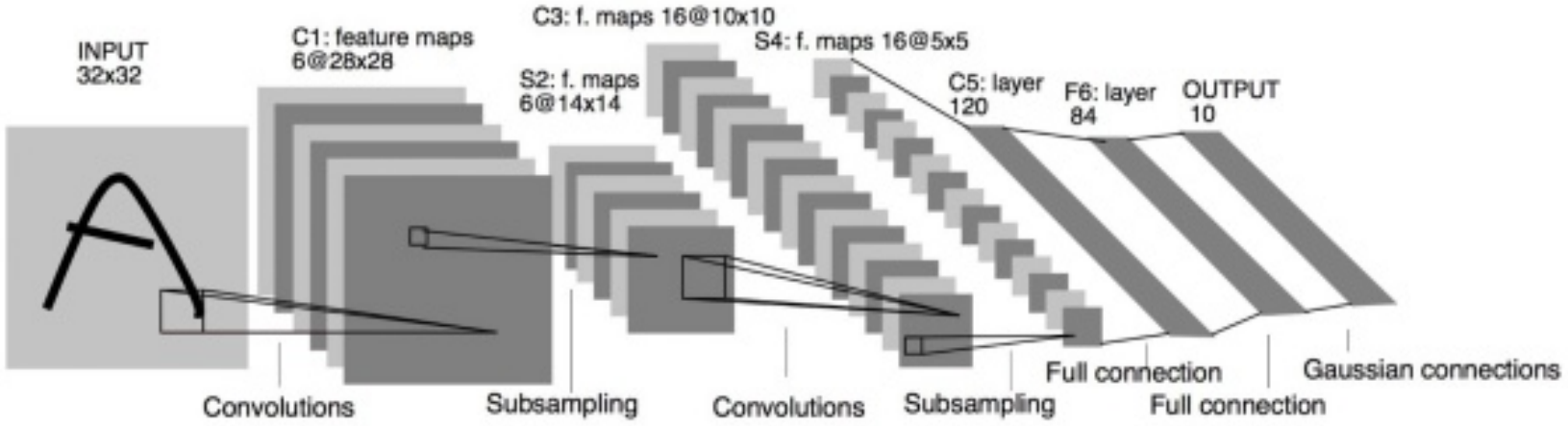
人工智慧與機器視覺

- CNN:卷積類神經網路
  - 卷積跟池化的計算會由很多人工神經元來處理
  - 而且是交互處理



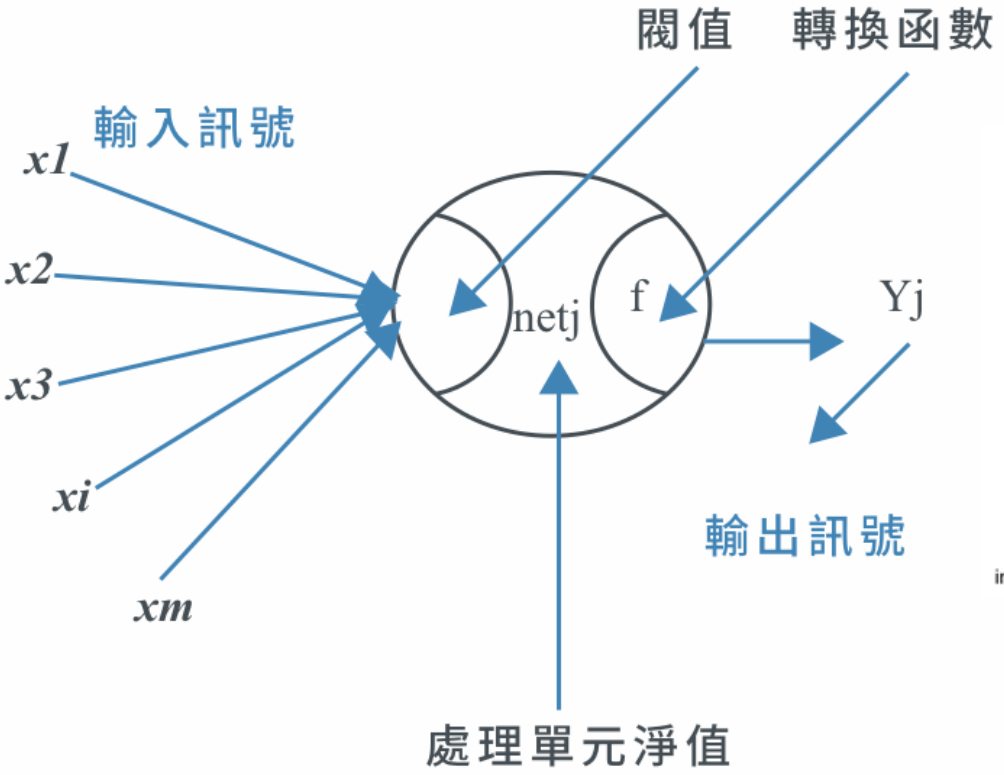
人工智慧
與
機器視覺

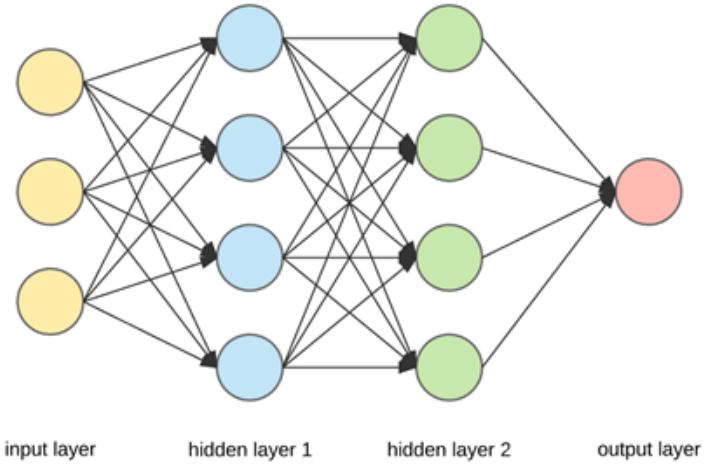
- CNN:卷積類神經網路示意圖



人工智慧
與
機器視覺

- 問題是
  - 到底什麼是機器的「學習」？
  - 學到什麼？





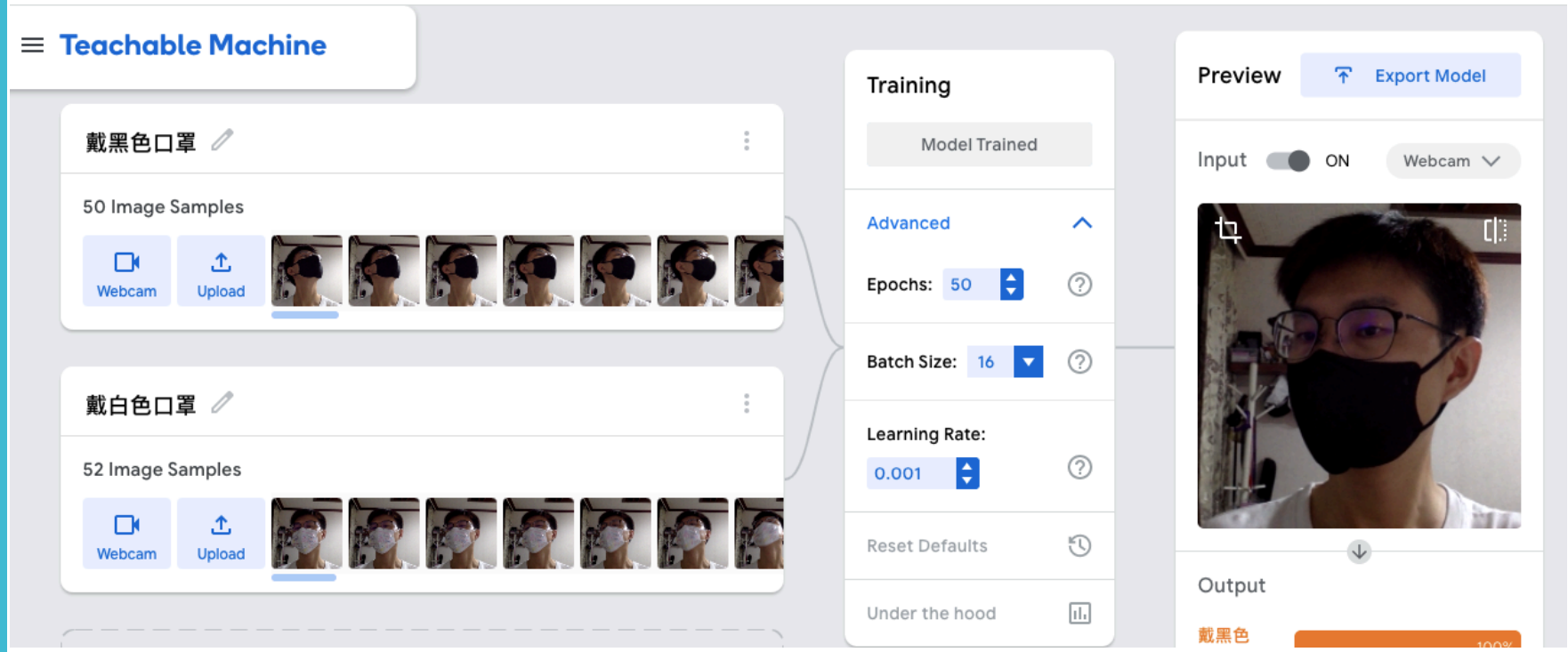
人工智慧
與
機器視覺

- 你做過的類似的東西：



人工智慧
與
機器視覺

- 你做過的類似的東西：



人工智慧
與
機器視覺

- 方便的CNN體驗：
  - https://www.cs.ryerson.ca/~aharley/vis/conv/

DonkeyCar與人工智慧

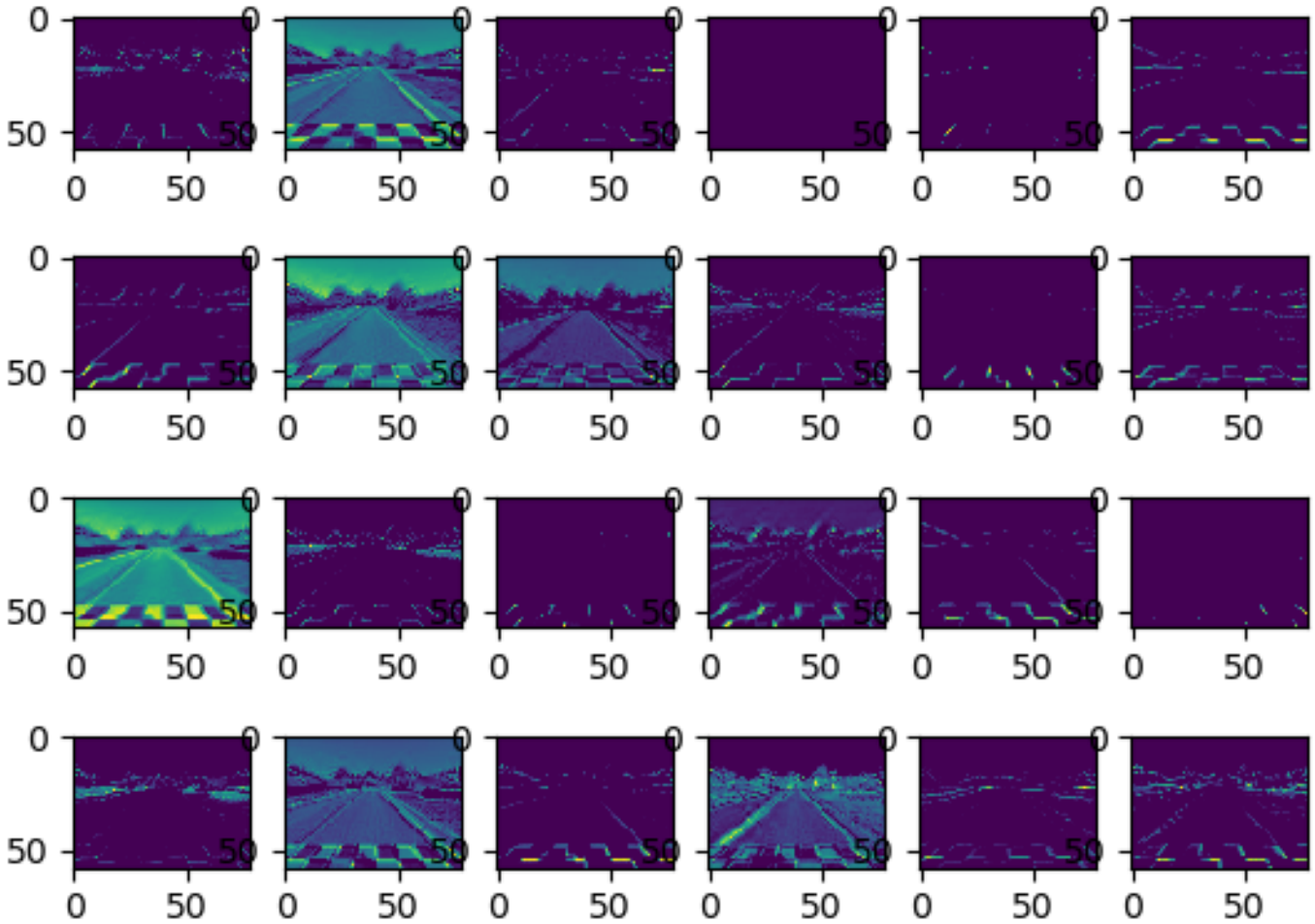
DonkeyCar 與人工智慧

- 原始圖片：
  - 160*120



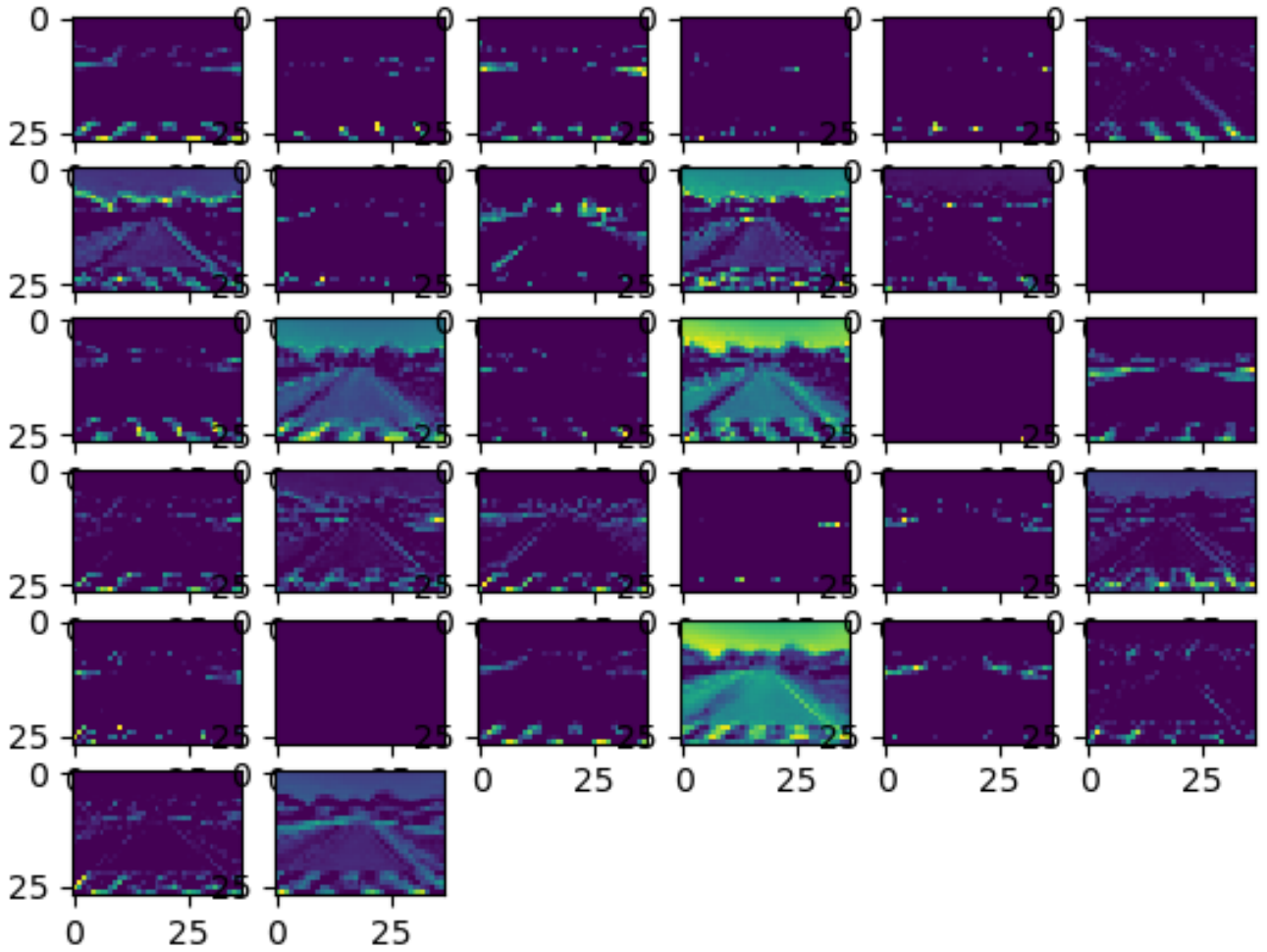
DonkeyCar 與人工智慧

- 處理後：
  - 第一層：24張



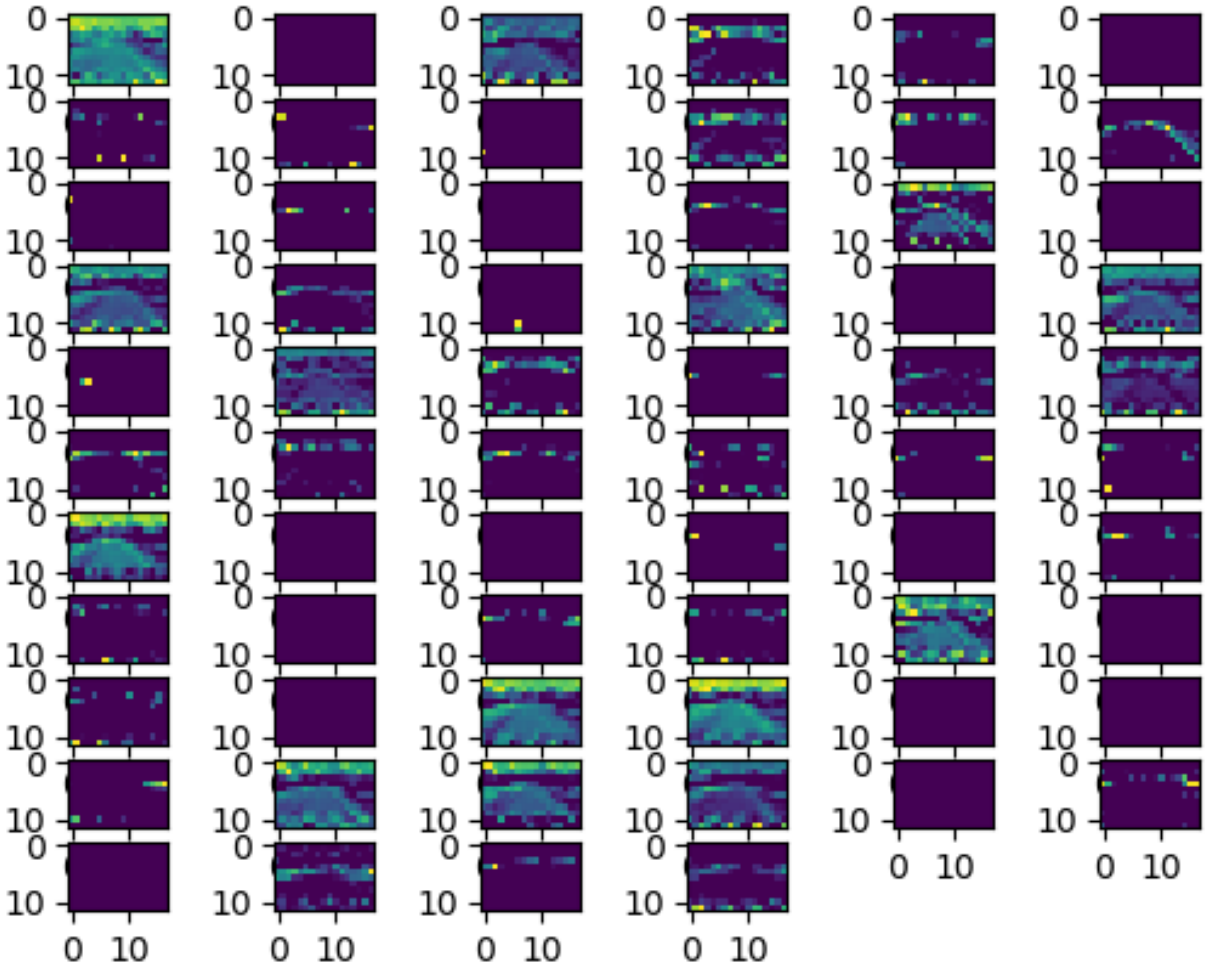
DonkeyCar 與人工智慧

- 處理後：
  - 第二層：32張



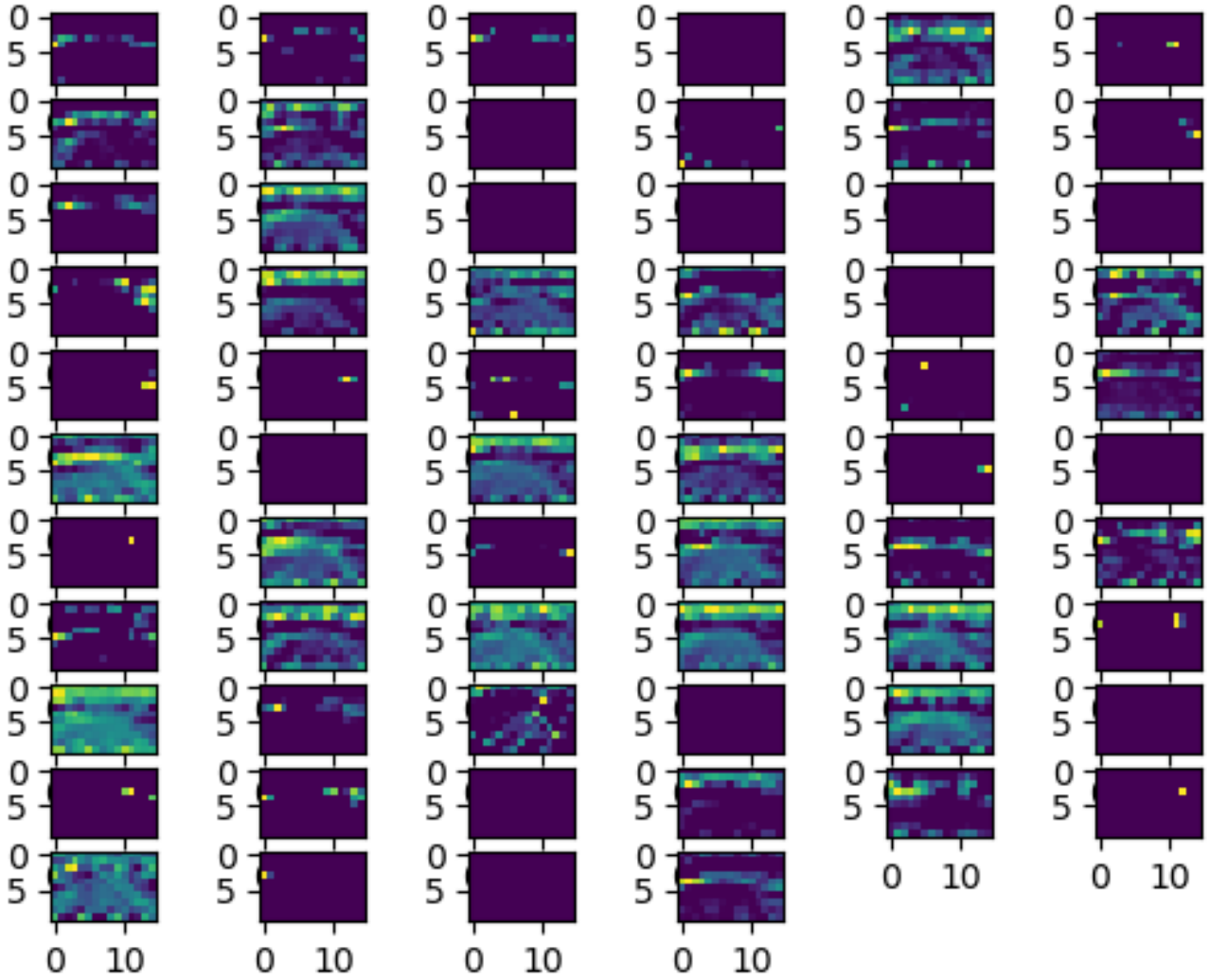
DonkeyCar 與人工智慧

- 處理後：
  - 第三層：64張



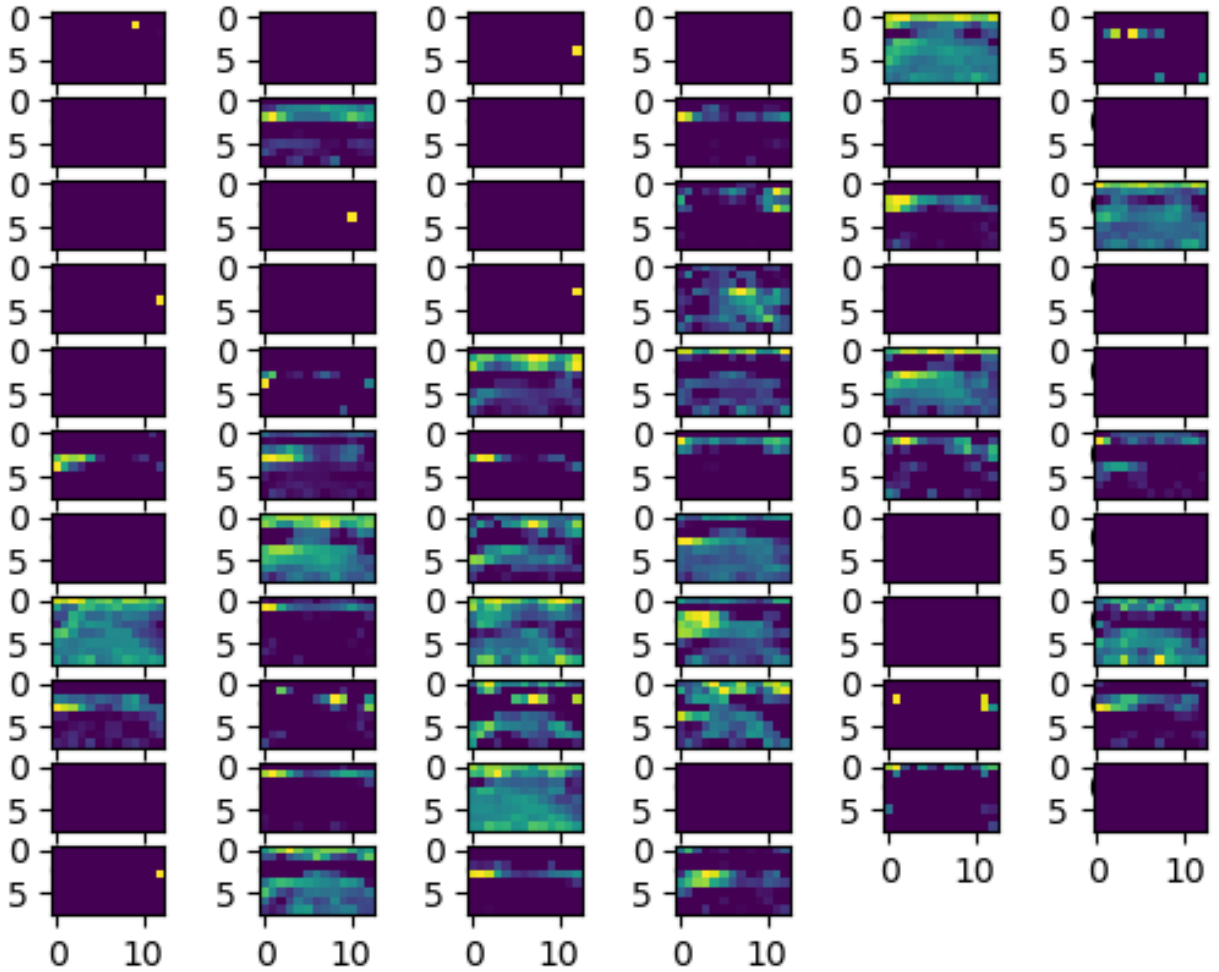
DonkeyCar 與人工智慧

- 處理後：
  - 第四層：64張



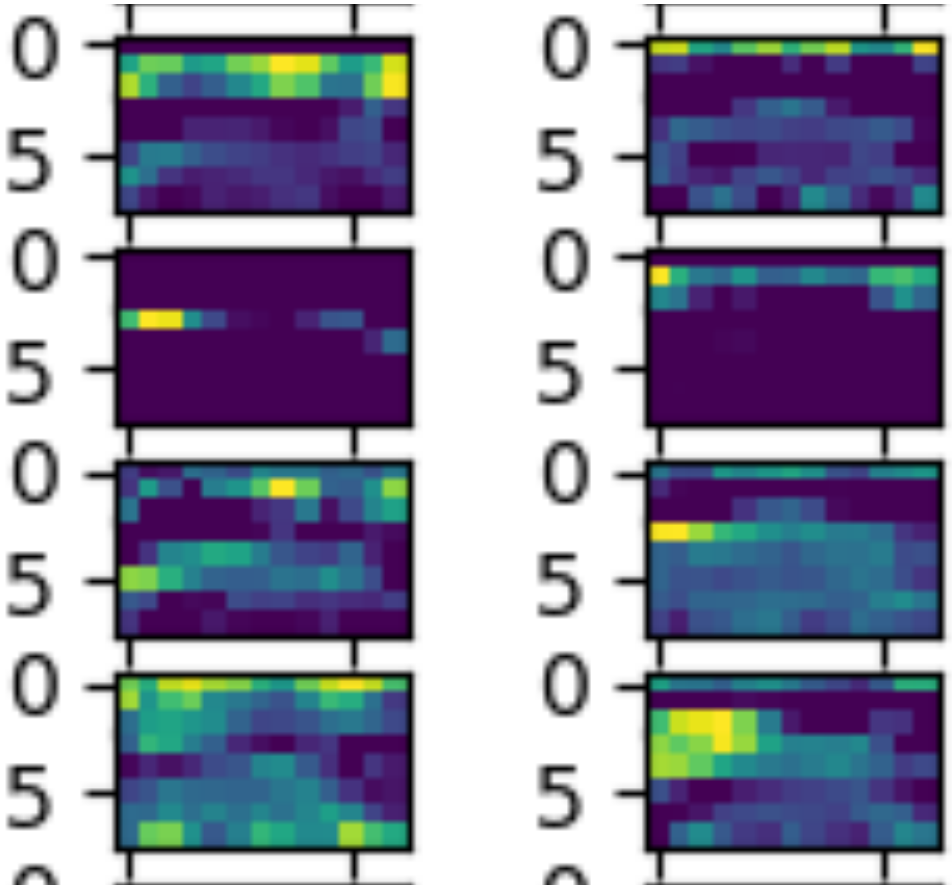
DonkeyCar 與人工智慧

- 處理後：
  - 第五層：64張



DonkeyCar 與人工智慧

- 處理後：
  - 第五層局部圖



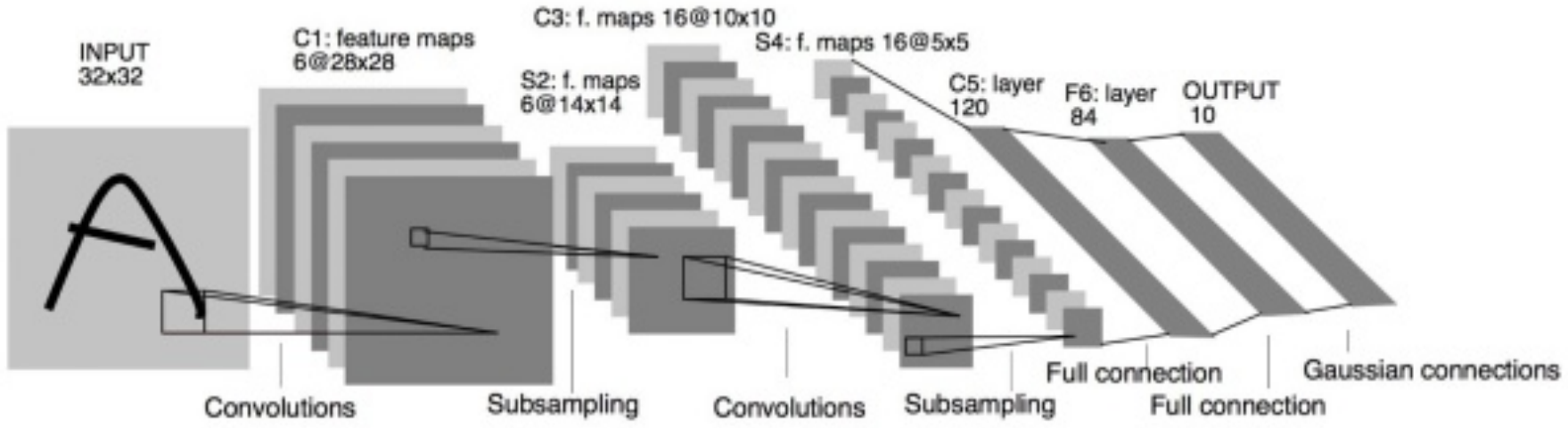
DonkeyCar 與人工智慧

- 數值：

```
layer 1 shape: (58, 78, 24)
layer 2 shape: (27, 37, 32)
layer 3 shape: (12, 17, 64)
layer 4 shape: (10, 15, 64)
layer 5 shape: (8, 13, 64)
```

DonkeyCar 與人工智慧

- CNN:卷積類神經網路示意圖

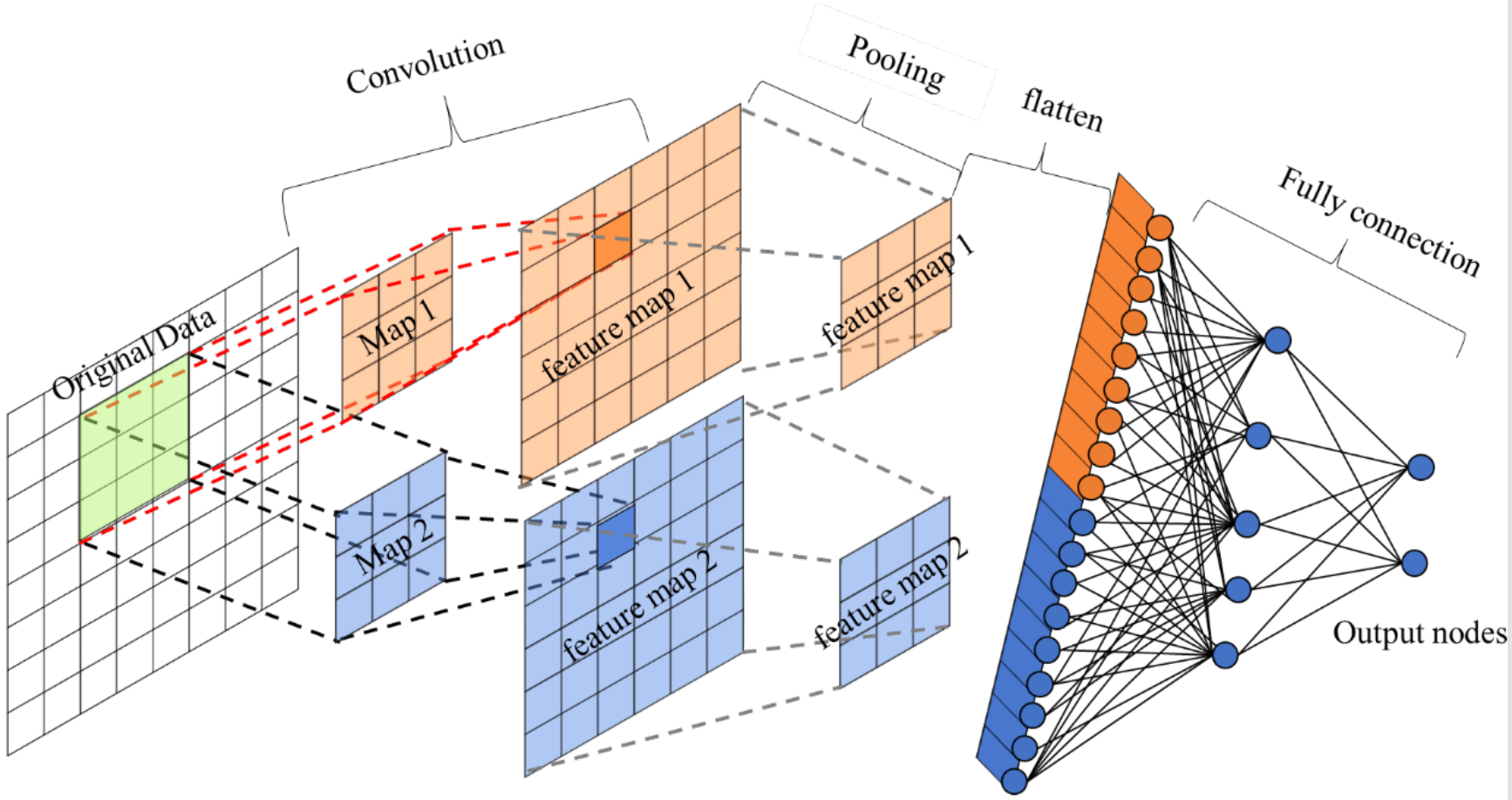


DonkeyCar 與人工智慧

- CNN:卷積類神經網路
  - 前面的特徵處理都是部分連接
  - 最後會有一個「全連接層」：把二維資料攤平成一維後輸入的神經元
  - 又稱為密集層（dense）
  - 在這裡，就是把每個特徵都用來做類別的判斷

DonkeyCar 與人工智慧

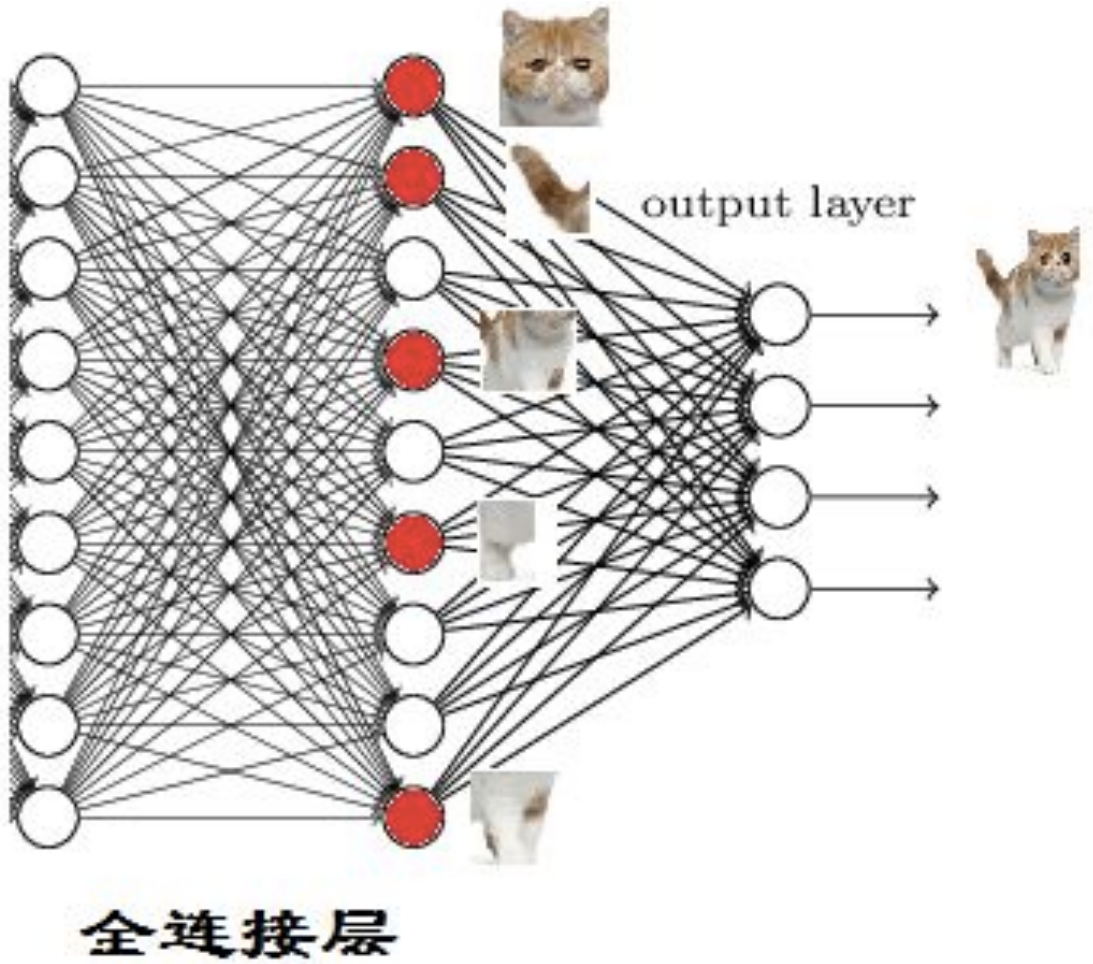
- CNN:卷積類神經網路示意圖



- 資料來源：https://medium.com/@chih.sheng.huang821/%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF-convolutional-neural-network-cnn-cnn%E9%81%8B%E7%AE%97%E6%B5%81%E7%A8%8B-ecaec240a631

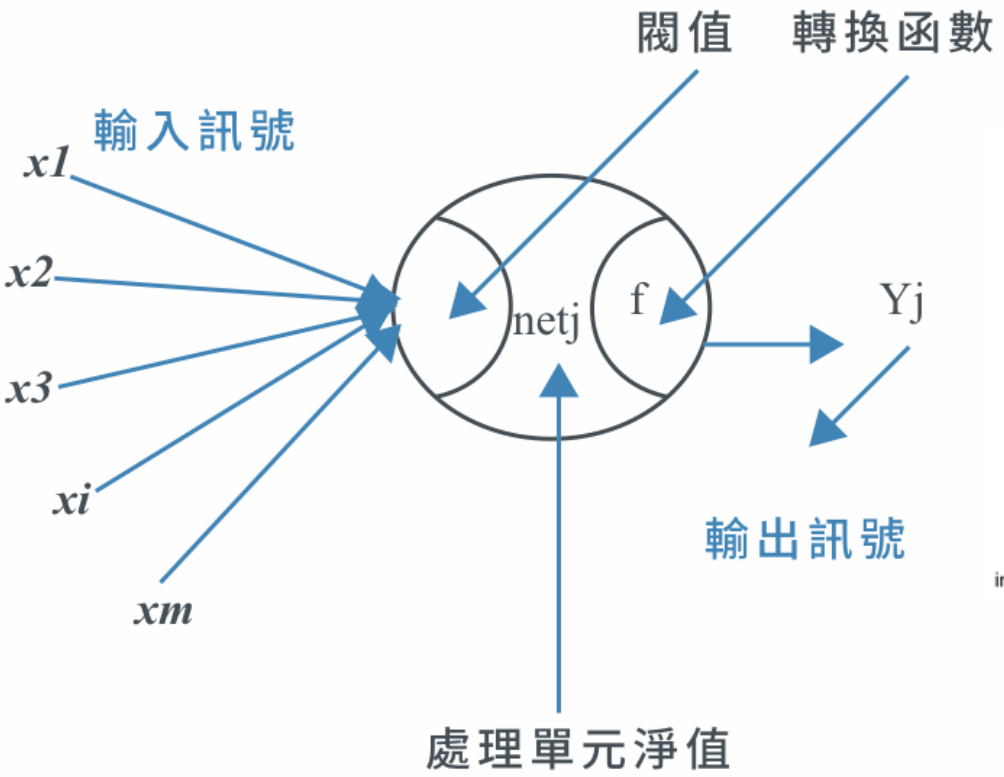
DonkeyCar 與人工智慧

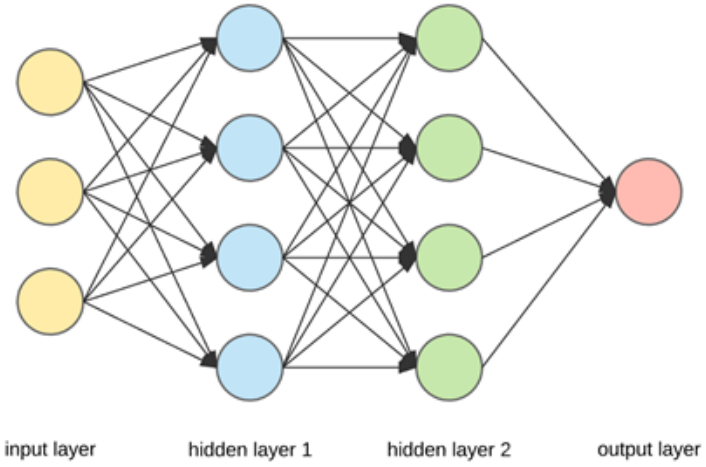
- CNN:卷積類神經網路
  - 輸出層



DonkeyCar 與人工智慧

- 問題是
  - 到底什麼是機器的「學習」？
  - 學到什麼？





DonkeyCar 與人工智慧

- DonkeyCar的學習

```
Epoch 4/100
16/16 [==============================] - 1s 45ms/step - loss: 0.3283 - n_outputs0_loss: 0

Epoch 00004: val_loss improved from 0.36250 to 0.32829, saving model to C:\projects\mysim
65/65 [==============================] - 12s 186ms/step - loss: 0.3497 - n_outputs0_loss:
loss: 0.0692
Epoch 5/100
16/16 [==============================] - 1s 45ms/step - loss: 0.2702 - n_outputs0_loss: 0

Epoch 00005: val_loss improved from 0.32829 to 0.27020, saving model to C:\projects\mysim
65/65 [==============================] - 12s 187ms/step - loss: 0.2954 - n_outputs0_loss:
loss: 0.0573
Epoch 6/100
16/16 [==============================] - 1s 46ms/step - loss: 0.2376 - n_outputs0_loss: 0

Epoch 00006: val_loss improved from 0.27020 to 0.23763, saving model to C:\projects\mysim
65/65 [==============================] - 12s 187ms/step - loss: 0.2567 - n_outputs0_loss:
loss: 0.0525
Epoch 7/100
16/16 [==============================] - 1s 46ms/step - loss: 0.2091 - n_outputs0_loss: 0

Epoch 00007: val_loss improved from 0.23763 to 0.20912, saving model to C:\projects\mysim
65/65 [==============================] - 12s 188ms/step - loss: 0.2119 - n_outputs0_loss:
loss: 0.0431
```

DonkeyCar 與人工智慧

- DonkeyCar的CNN

```
training with model type <class 'donkeycar.parts.keras.KerasLinear'>
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
=================================================================
img_in (InputLayer)          [(None, 224, 224, 3) 0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 110, 110, 24) 1824      img_in[0][0]
_________________________________________________________________
dropout (Dropout)            (None, 110, 110, 24) 0         conv2d_1[0][0]
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 53, 53, 32)   19232     dropout[0][0]
_________________________________________________________________
dropout_1 (Dropout)          (None, 53, 53, 32)   0         conv2d_2[0][0]
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 25, 25, 64)   51264     dropout_1[0][0]
_________________________________________________________________
dropout_2 (Dropout)          (None, 25, 25, 64)   0         conv2d_3[0][0]
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 23, 23, 64)   36928     dropout_2[0][0]
_________________________________________________________________
```

System throttled

DonkeyCar 與人工智慧

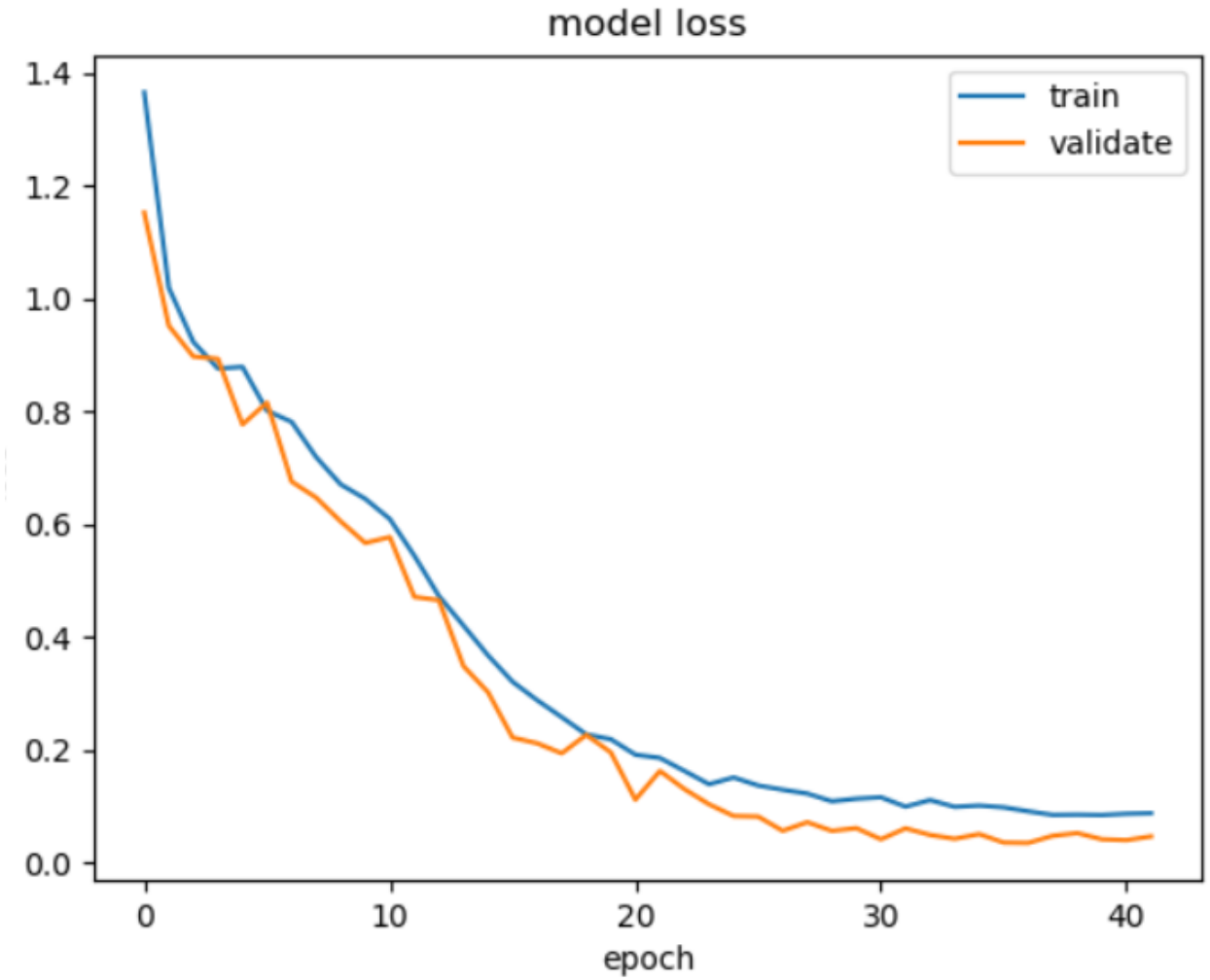
- DonkeyCar的CNN

```
dropout_3 (Dropout)          (None, 23, 23, 64)    0         conv2d_4[0][0]
conv2d_5 (Conv2D)            (None, 21, 21, 64)    36928     dropout_3[0][0]
dropout_4 (Dropout)          (None, 21, 21, 64)    0         co
flattened (Flatten)          (None, 28224)         0         dropout_4[0][0]
dense (Dense)                (None, 100)           2822500   flattened[0][0]
dropout_5 (Dropout)          (None, 100)           0         dense[0][0]
dense_1 (Dense)              (None, 50)            5050      dropout_5[0][0]
dropout_6 (Dropout)          (None, 50)            0         dense_1[0][0]
n_outputs0 (Dense)           (None, 1)             51        dropout_6[0][0]
n_outputs1 (Dense)           (None, 1)             51        dropout_6[0][0]
==================================================================================
Total params: 2,973,828
Trainable params: 2,973,828
Non-trainable params: 0
```

System throttled due to Ov

DonkeyCar 與人工智慧

- 收斂圖：



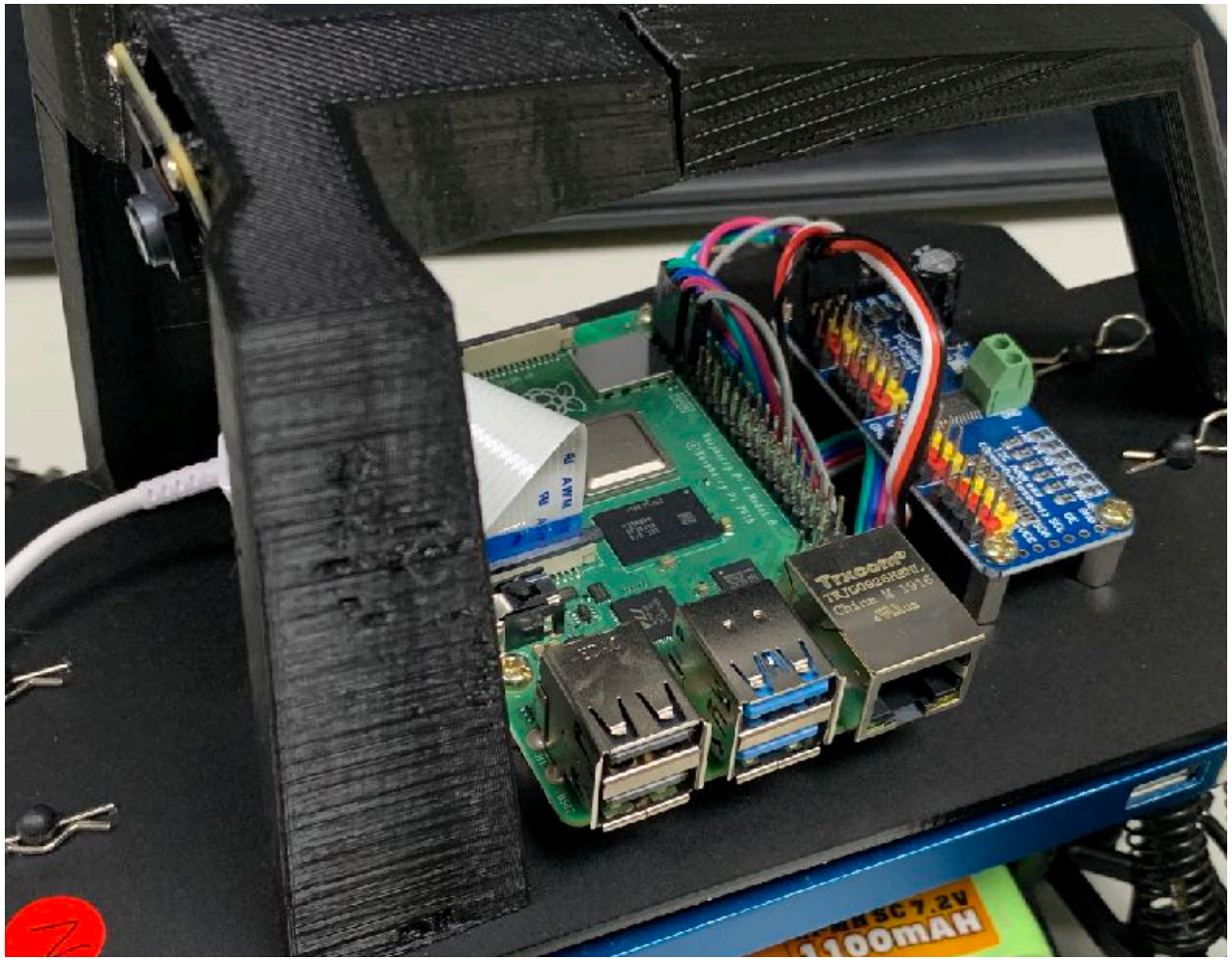
DonkeyCar 與人工智慧

- 快速上手的CNN體驗：
  - https://poloclub.github.io/cnn-explainer/
  - https://www.cs.ryerson.ca/~aharley/vis/conv/
- 更多種神經網路的連接方式：
  - https://kknews.cc/zh-tw/finance/393eo8g.html

實體自駕車

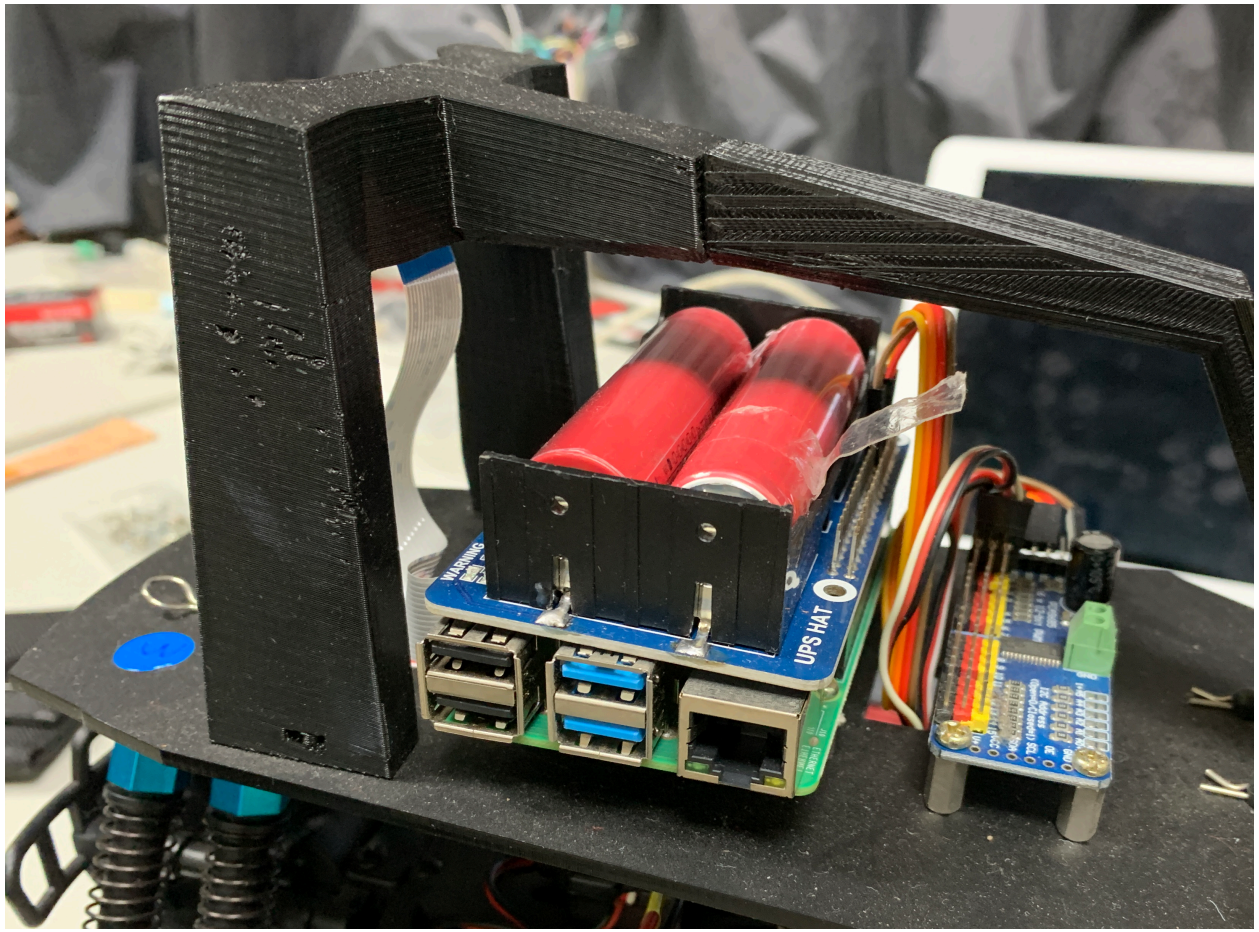
實體自駕車

- 核心：樹莓派



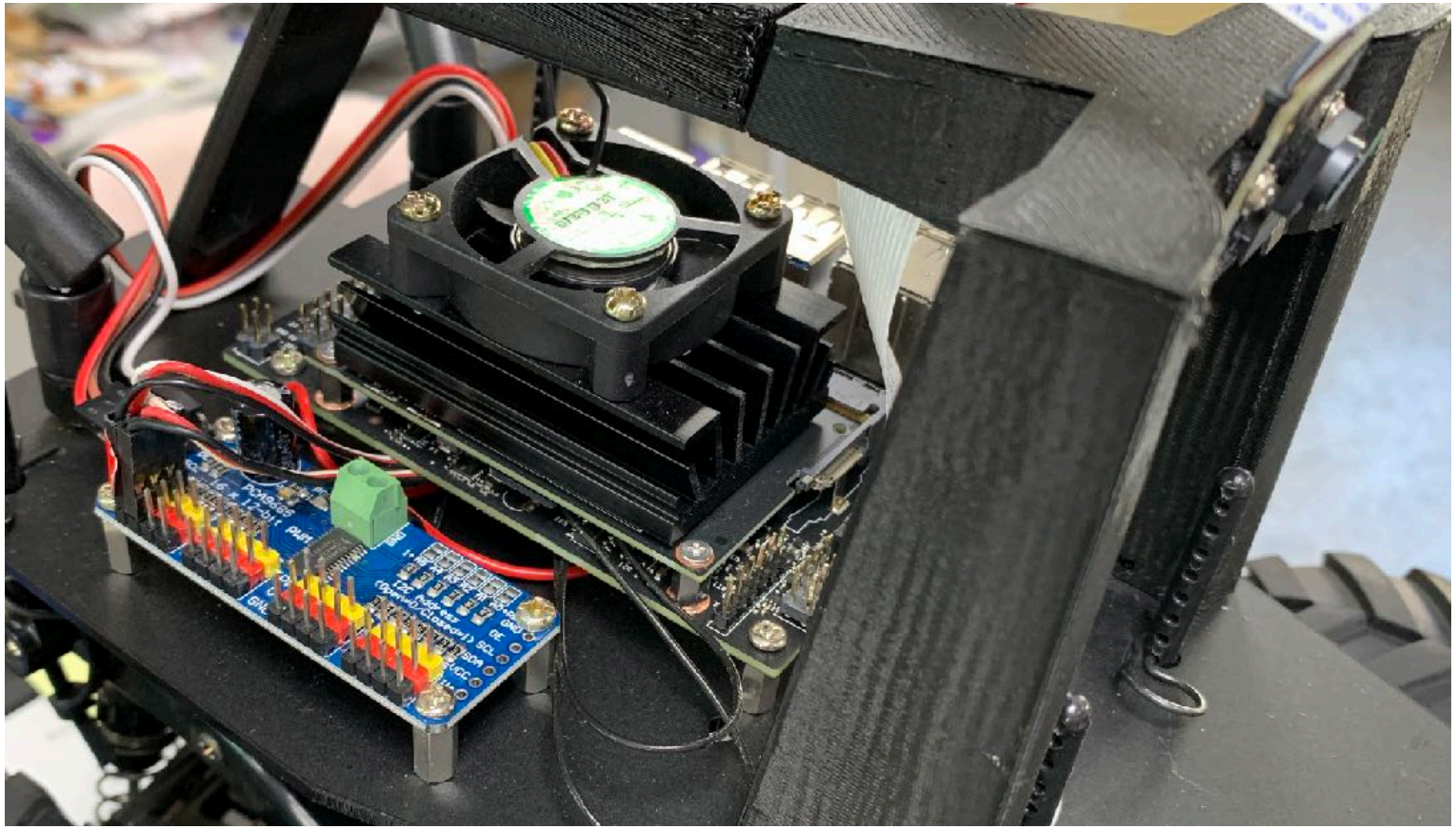
實體自駕車

- 核心：樹莓派＋電源



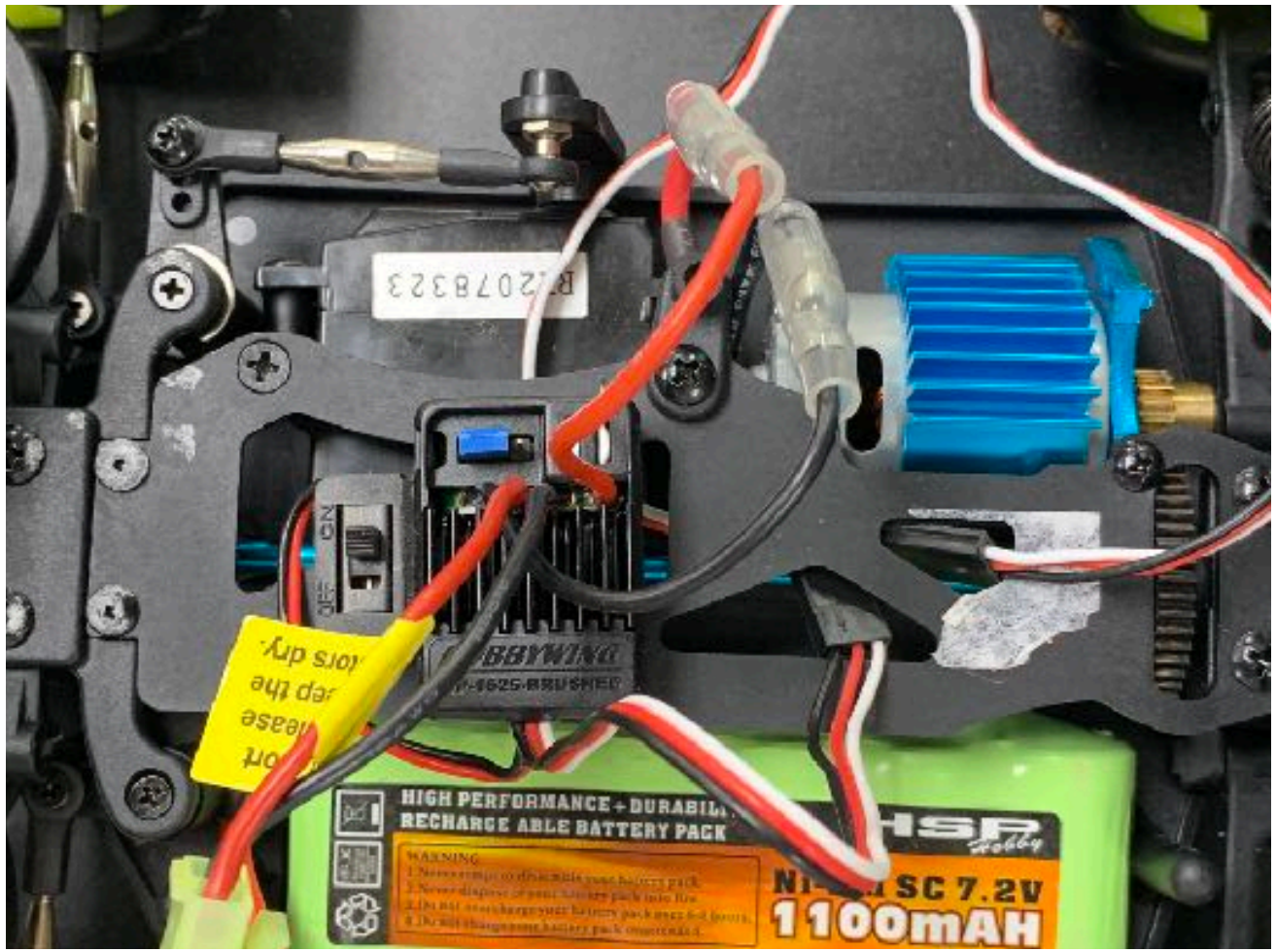
實體自駕車

- 核心：JetsonNano



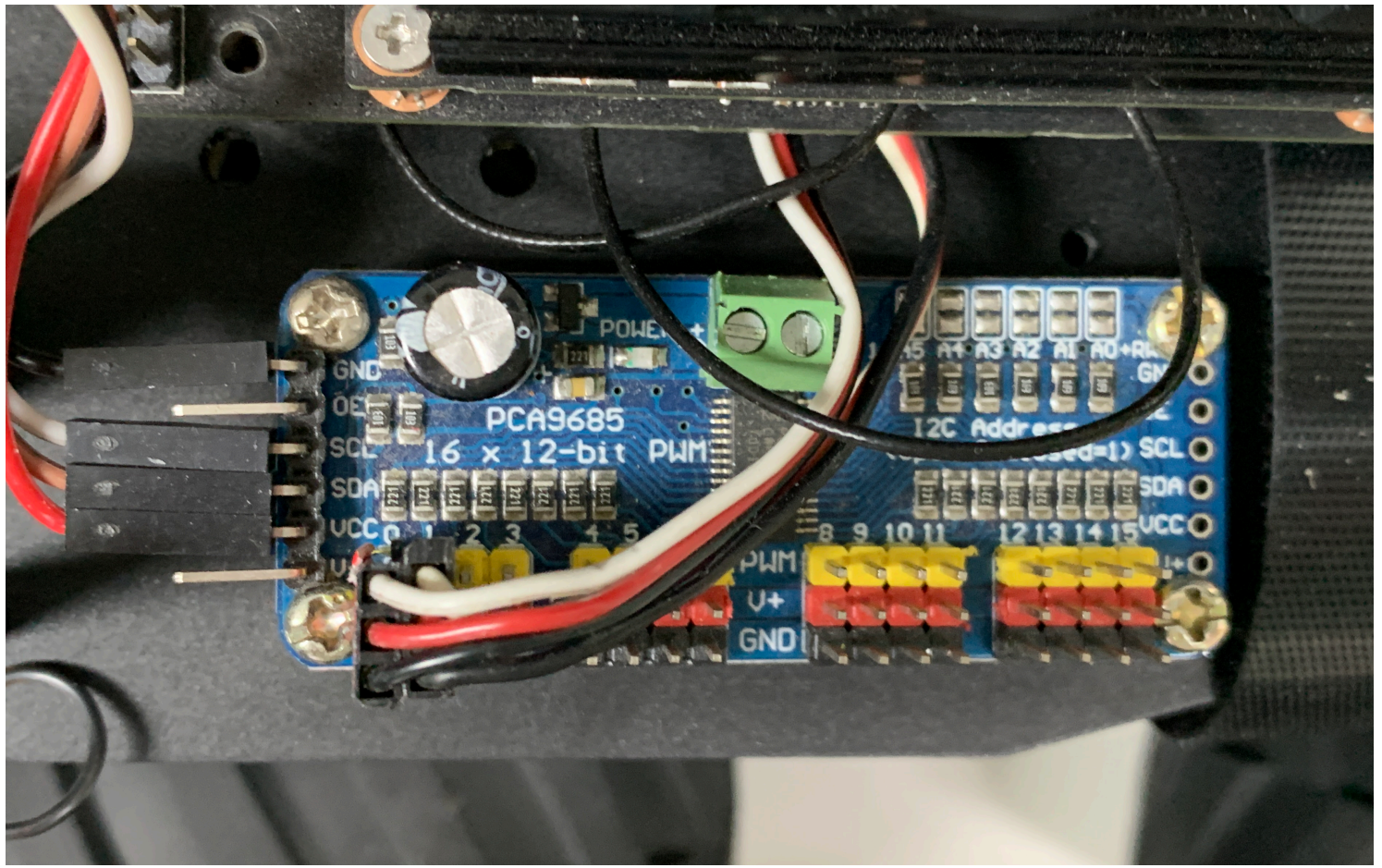
實體自駕車

- 碳刷馬達與電源



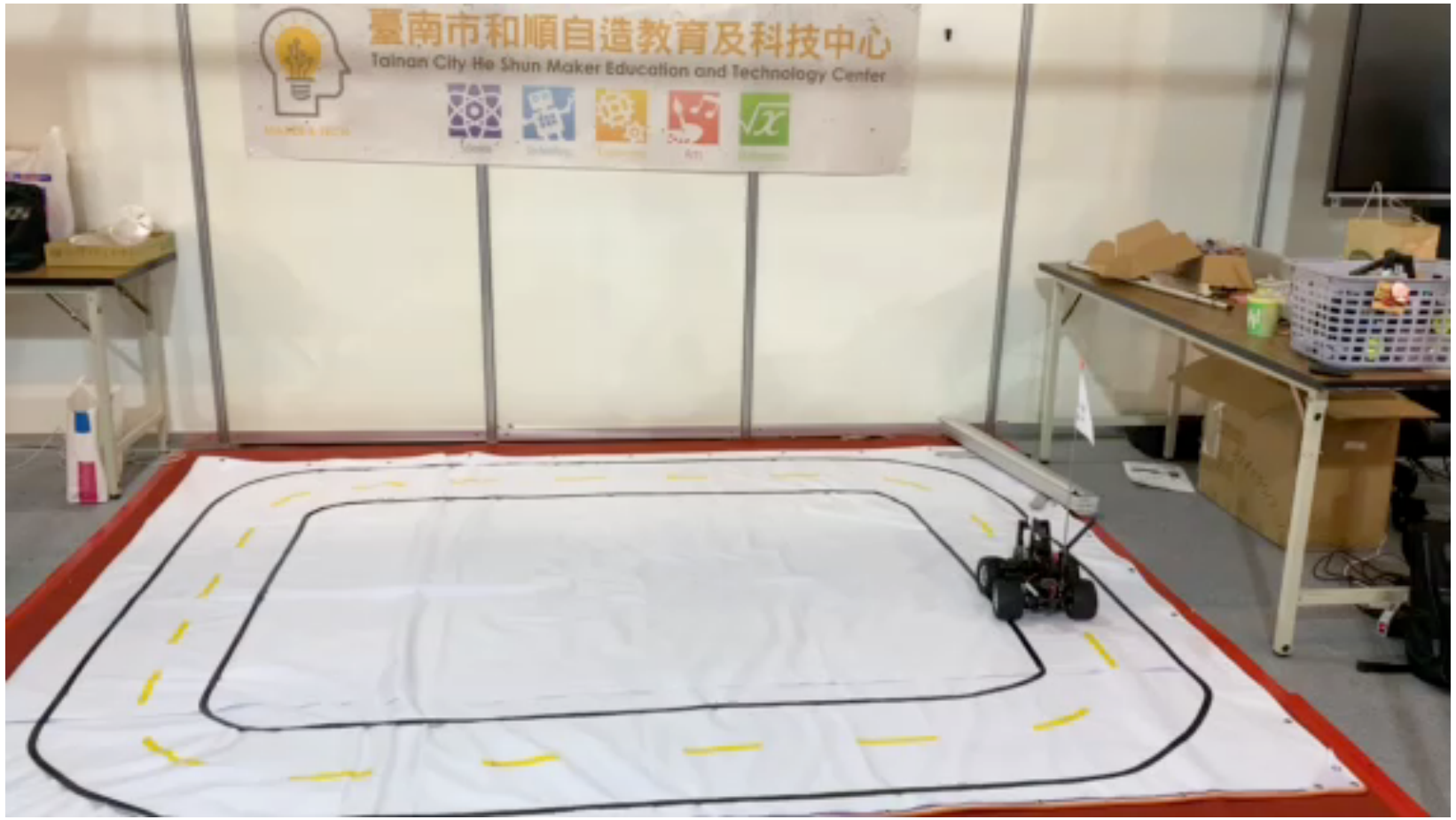
實體自駕車

- 馬達控制板：



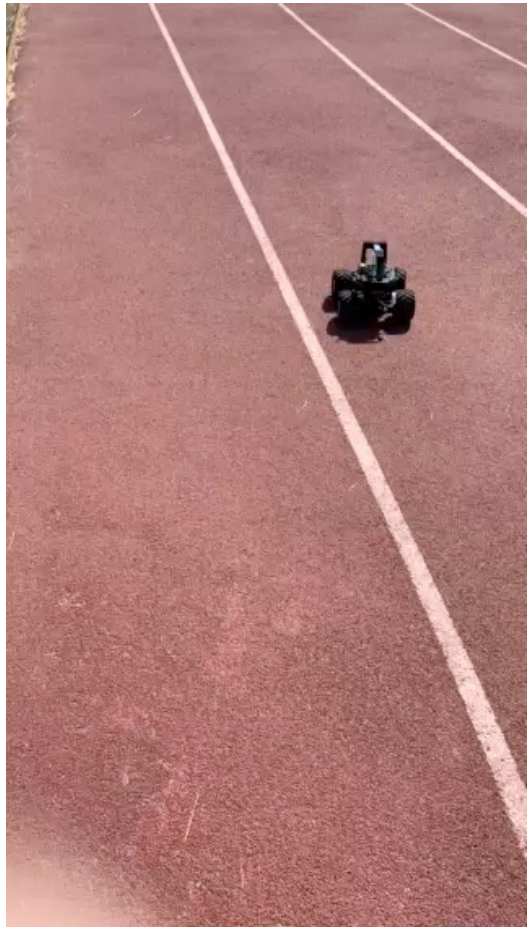
實體自駕車

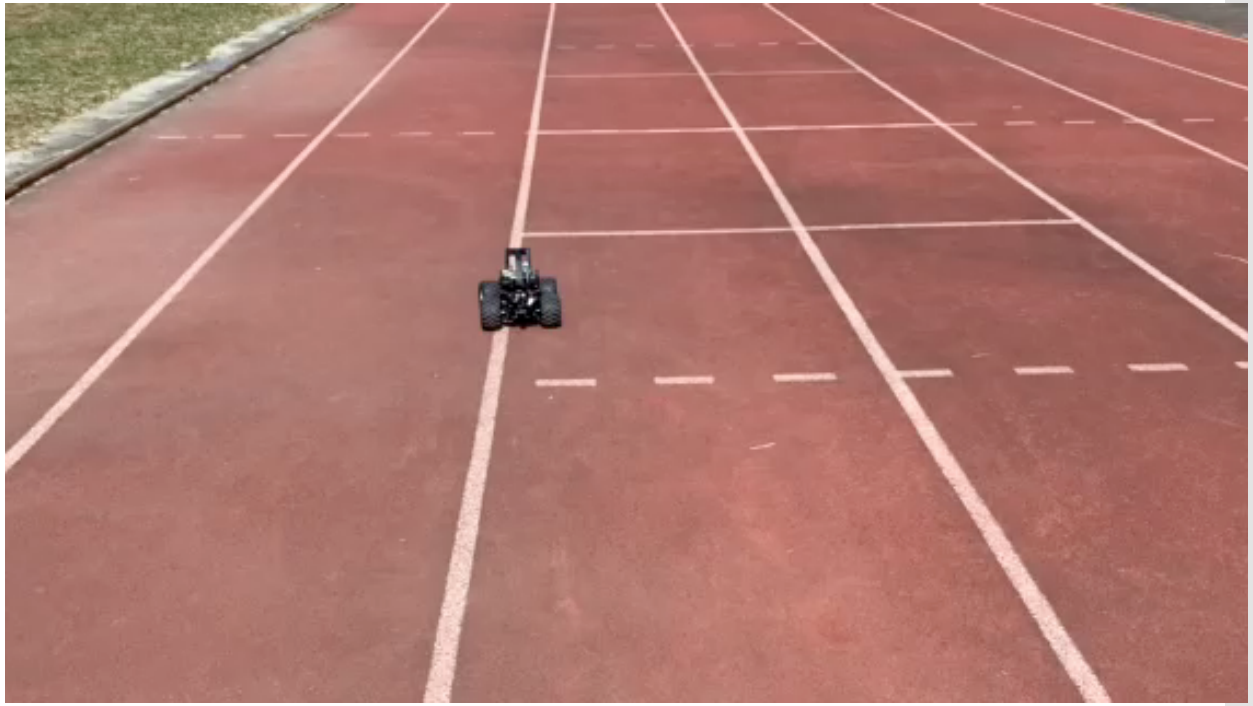
- 訓練結果：


臺南市和順自造教育及科技中心
Tainan City He Shun Maker Education and Technology Center

實體自駕車

- 訓練結果：





數學時間

數學時間

- 數值有什麼秘密🈯？
- 輸入是120*160

```
layer 1 shape: (58, 78, 24)
layer 2 shape: (27, 37, 32)
layer 3 shape: (12, 17, 64)
layer 4 shape: (10, 15, 64)
layer 5 shape: (8, 13, 64)
```

數學時間

- 你看得出來kernel是幾＊幾嗎？

```
training with model type <class 'donkeycar.parts.keras.KerasLinear'>
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
img_in (InputLayer)          [(None, 224, 224, 3)      0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 110, 110, 24)      1824      
_________________________________________________________________
dropout (Dropout)            (None, 110, 110, 24)      0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 53, 53, 32)        19232     
_________________________________________________________________
dropout_1 (Dropout)          (None, 53, 53, 32)        0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 25, 25, 64)        51264     
_________________________________________________________________
dropout_2 (Dropout)          (None, 25, 25, 64)        0         
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 23, 23, 64)        36928     
_________________________________________________________________
```

System throttled

數學時間

- 參數量怎麼來的？

dropout_3 (Dropout)	(None, 23, 23, 64)	0	conv2d_4[0][0]
conv2d_5 (Conv2D)	(None, 21, 21, 64)	36928	dropout_3[0][0]
dropout_4 (Dropout)	(None, 21, 21, 64)	0	co
flattened (Flatten)	(None, 28224)	0	dropout_4[0][0]
dense (Dense)	(None, 100)	2822500	flattened[0][0]
dropout_5 (Dropout)	(None, 100)	0	dense[0][0]
dense_1 (Dense)	(None, 50)	5050	dropout_5[0][0]
dropout_6 (Dropout)	(None, 50)	0	dense_1[0][0]
n_outputs0 (Dense)	(None, 1)	51	dropout_6[0][0]
n_outputs1 (Dense)	(None, 1)	51	dropout_6[0][0]

Total params: 2,973,828
Trainable params: 2,973,828
Non-trainable params: 0

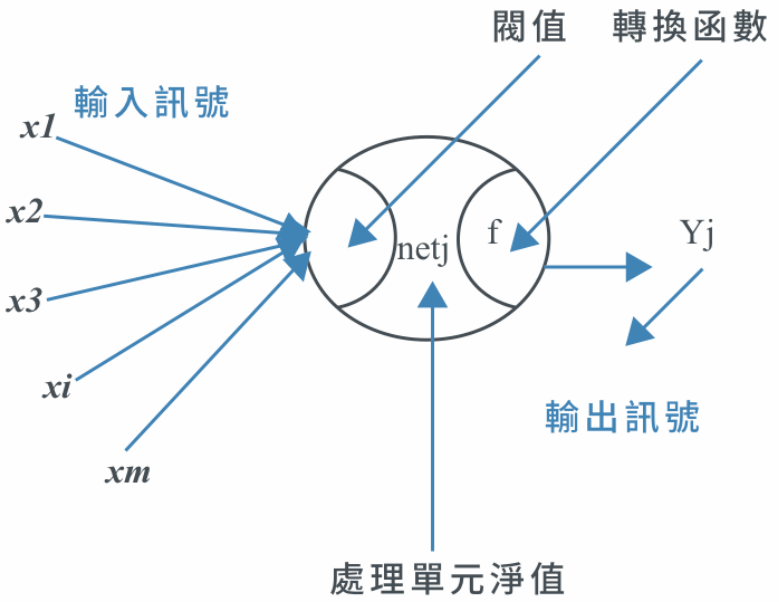
System throttled due to Ov

數學時間

- 耗費多少記憶體？
- https://blog.csdn.net/SweetSeven_/article/details/96026706

數學時間

- 神經元的數量？
  - 看輸出的數量？
  - 看參數的數量？
  - 看kernel的數量？
  - 看kernal掃過的數量？



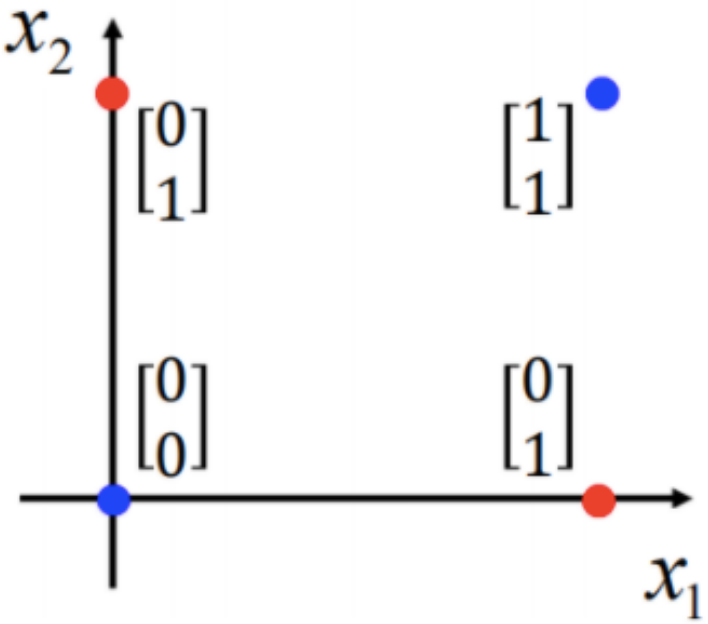
數學時間

- 為何CNN模型常有兩個全連接層？
- CNN第一次的filter在做什麼？那第二次以後呢？
- 每個神經元的激勵函數（activation function）在幹嘛？

數學時間

- 什麼叫做非線性分類？

Input Feature		Label
x_1	x_2	
0	0	Class 2
0	1	Class 1
1	0	Class 1
1	1	Class 2



數學時間

- 獻給喜歡數學的你❤️
- https://chih-sheng-huang821.medium.com/機器學習-神經網路-多層感知機-multilayer-perceptron-mlp-含詳細推導-ee4f3d5d1b41
- https://chih-sheng-huang821.medium.com/機器-統計學習-羅吉斯回歸-logistic-regression-aff7a830fb5d

Thank You

oceanblue0212@hsjh.tn.edu.tw