



Flower Classification

資工二 109321005 楊樸霖
資工二 109321020 沈奕菱
資工二 109321041 劉彥汝

目錄

01

動機及目的-----p3

02

模型建置說明-----p6

03

比較與分析-----p12

04

分工及時程表-----p19

05

遇到的困難-----p22

06

參考資料-----p24

01

The first chapter

動機及目的

希望女友大人開心

動機

送花的動機

根據心理學家的研究，花在打動女人芳心有著不可動搖的地位，不論是浪漫或是現實主義的女人，收到**心儀的對象送的花**，都會發自內心的感到喜悅



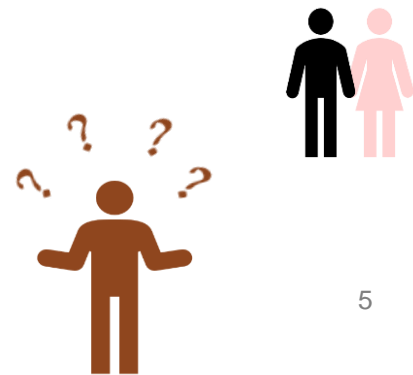
送花參考原則：

1. 花要漂亮，但不能比女友漂亮
2. 花語要優美能打動人心 (注意花種跟顏色)
3. 花的排列組合，如何能做出最完美的作品
4. **女友喜歡**



目的

每一朵相似外表的花，卻各有特點與美麗之處。
這份作業的宗旨是讓大家更加認識不同品種的花卉。



02

The sencod chapter

模型建置說明

資料數量、規格、前處理...模型架構、參數設定

Flowers Chosen(Each amount in our data set)

Astilbe
(740)



Bellflower
(886)



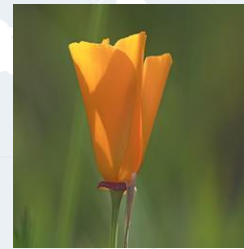
Black-eyed
susan(1000)



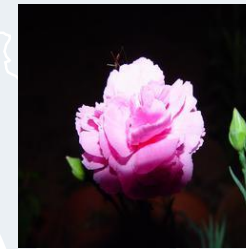
Calendula
(1025)



California
poppy(1035)



Carnation
(938)



Common
daisy(992)



Coreopsis
(1049)



Dandelion
(1052)



Iris
(1055)



Rose
(1000)



Sunflower
(1027)



Tulip
(1048)



Water lily
(991)



資料規格

尺寸

- 長*寬：256*256
- 位元深度：24

顏色

- 色彩

解析度

- 水平：96dip
- 垂直：96dip

格式

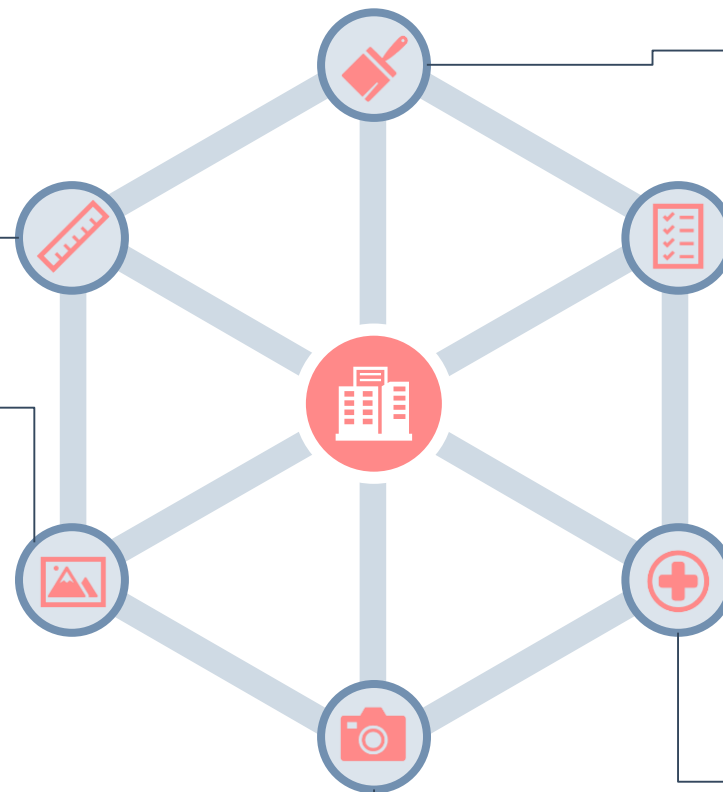
- JPG檔

焦距

- 35mm

大小

- 15KB~20KB



前處理

資料分類

```
splitfolders.ratio('flowers', output="output", seed=101, ratio=(0.8, 0.1, 0.1))
```

- 分成三堆，train、valivation、test，比例為 8 : 1 : 1

圖像增強及正規化

- **rescale**：將畫素值縮放到0~1區間
- **rotation_range**：角度值0~180，影象旋轉
- **width_shift_range**：水平平移，相對總寬度的比例
- **height_shift_range**：垂直平移，相對總高度的比例
- **shear_range**：隨機錯切角度
- **zoom_range**：隨機縮放範圍
- **horizontal_flip**：一半影象水平翻轉
- **fill_mode**：填充新建立畫素的方法

```
train_datagen = ImageDataGenerator(rescale = 1.0/255.0,  
                                   rotation_range = 0.5,  
                                   width_shift_range = 0.2,  
                                   height_shift_range = 0.2,  
                                   shear_range = 0.2,  
                                   zoom_range = 0.1,  
                                   horizontal_flip = True,  
                                   fill_mode = 'nearest'  
                                   )  
  
test_val_datagen = ImageDataGenerator(rescale = 1.0/255.0)
```

改變圖片大小

- **batch_size**：一批次有幾個樣本
- **img_width**：照片寬度(左-右)
- **img_height**：照片長度(上-下)

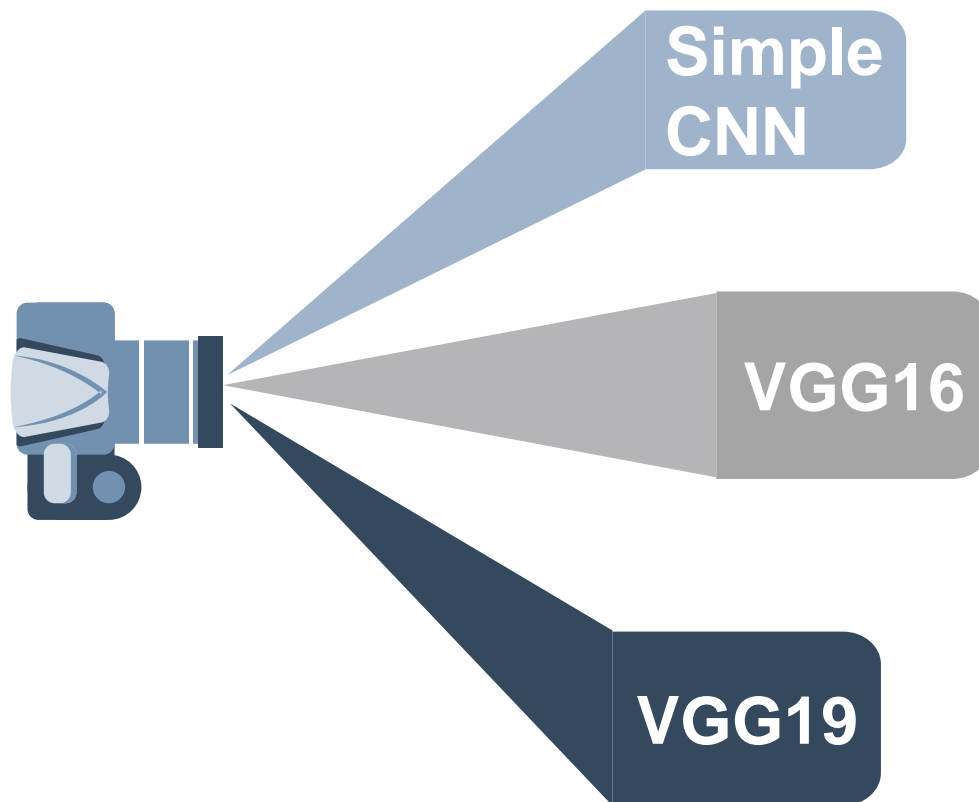
```
batch_size = 64  
img_width = 128  
img_height = 128
```

所使用的模型架構器

模型架構：

Simple CNN為我們組自己寫出的一套模型架構。

VGG為Visual Geometry Group的縮寫，提供許多隱藏層與大量圖片訓練，提高準確率至90%。



16層，
13卷積層與
3個全連接層

19層，
16卷積層與
3個全連接層

架構

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 128, 3)]	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 13)	106509
Total params: 14,821,197 Trainable params: 106,509 Non-trainable params: 14,714,688		

Model: "model"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 128, 3)]	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590080
block3_conv4 (Conv2D)	(None, 32, 32, 256)	590080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv4 (Conv2D)	(None, 16, 16, 512)	2359808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv4 (Conv2D)	(None, 8, 8, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 13)	106509
Total params: 20,130,893 Trainable params: 106,509 Non-trainable params: 20,024,384		

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 16)	448
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_3 (Conv2D)	(None, 16, 16, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 128)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 128)	1048704
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 13)	1677
Total params: 1,147,821 Trainable params: 1,147,821 Non-trainable params: 0		



03

The third chapter

比較與分析

模型效率評估

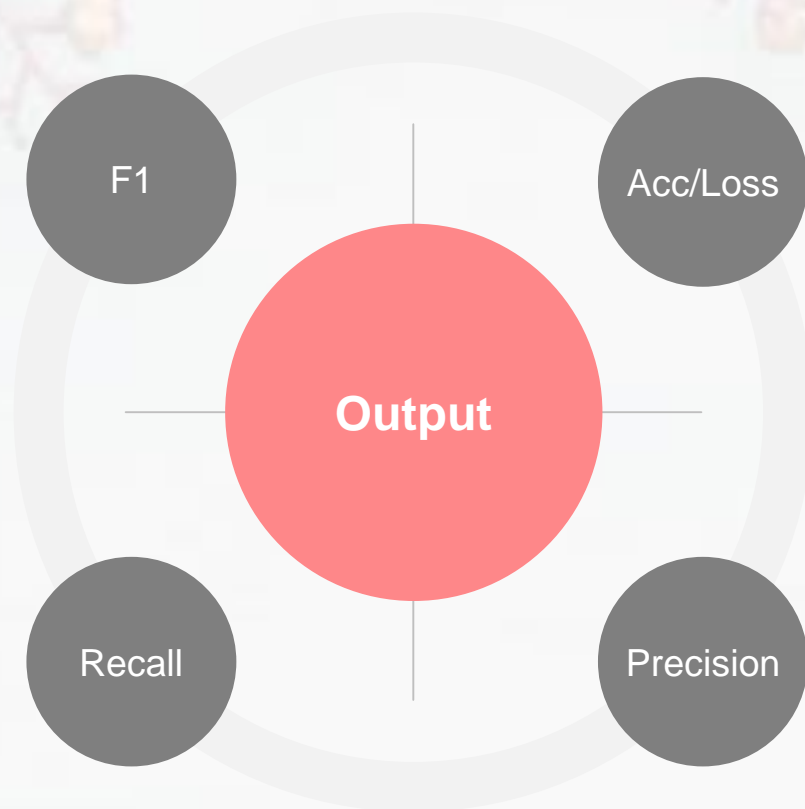
測試結果簡介

F1-score

F1-score是兩者的調和平均數，算是一個比較概略的指標來看這個模型的表現

Recall

Recall是在所有正樣本當中，能夠預測多少正樣本的比例



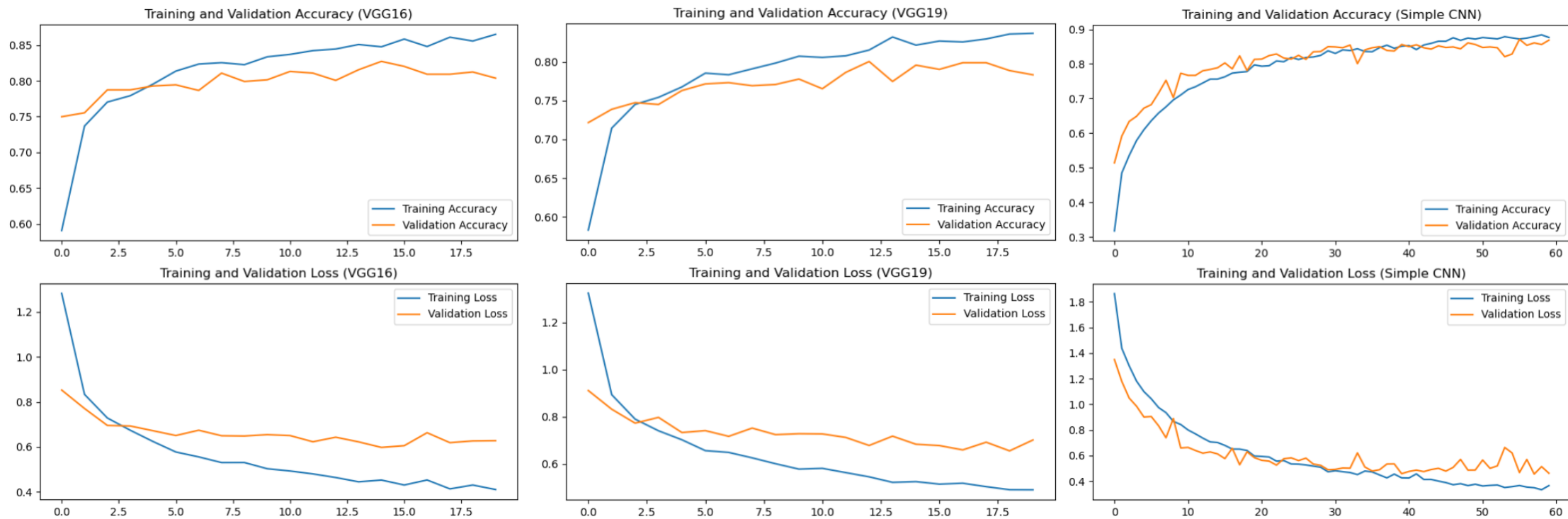
Accuracy/Loss

準確性與失誤率，最直白的效率好壞辨別標準

Precision

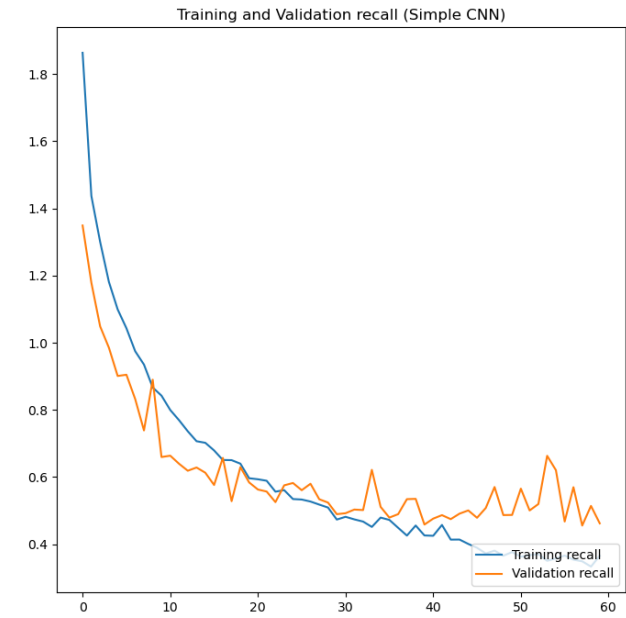
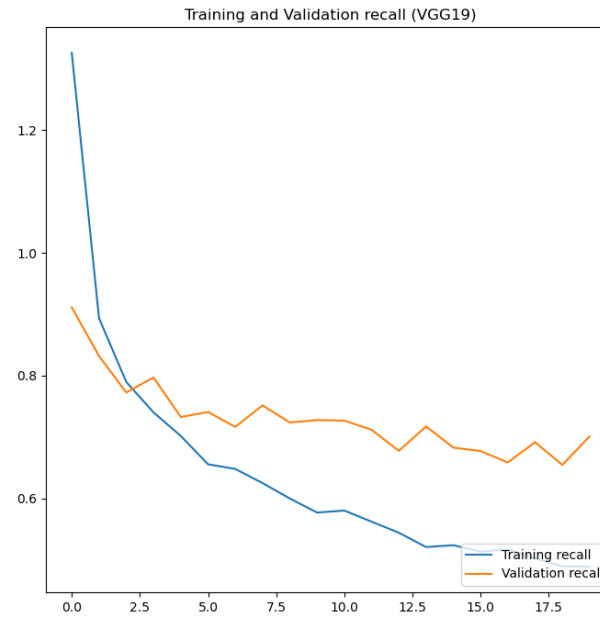
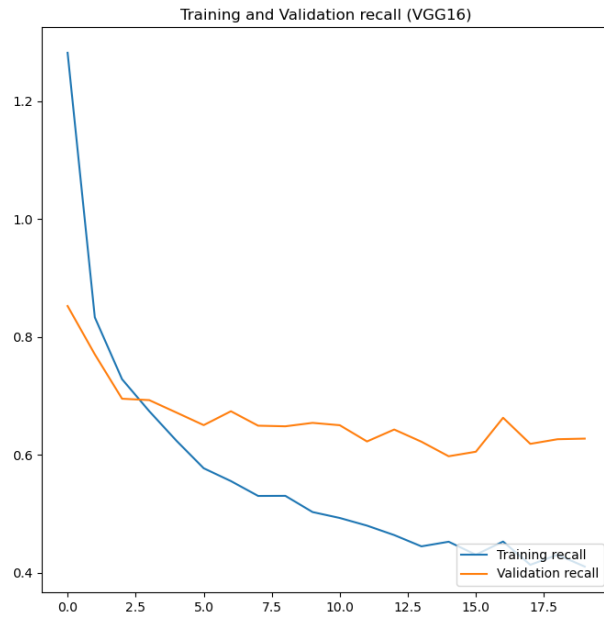
Precision為在所有預測為正樣本中，有多少為正樣本。

Accuracy and Loss

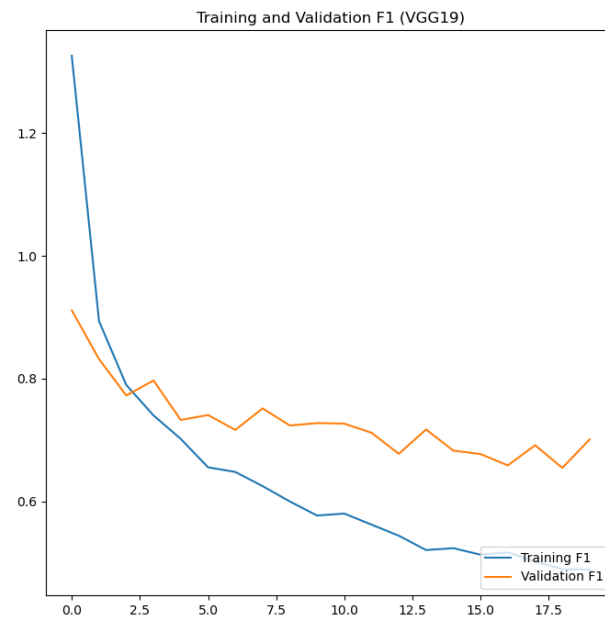
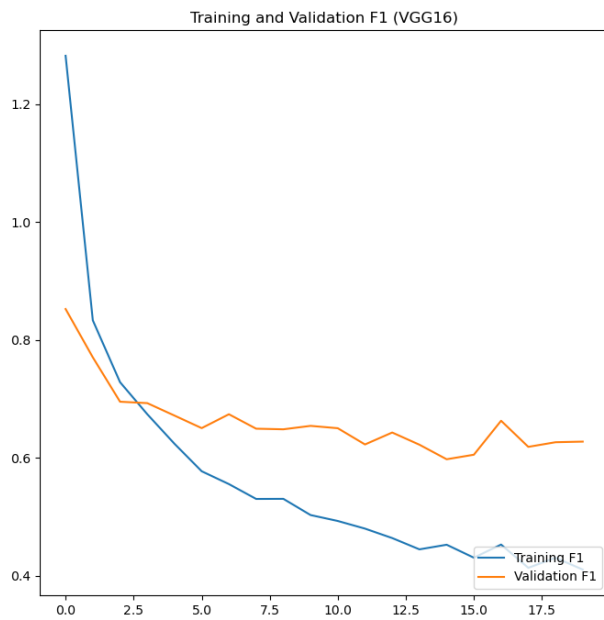


- 可觀察到VGG架構所設定epoch之數目沒有Simple CNN多，但訓練出來的accuracy及loss卻與CNN有著差不多的水準，可見我們這次訓練出來的模型可以跟網路上熱門模型互相匹敵。

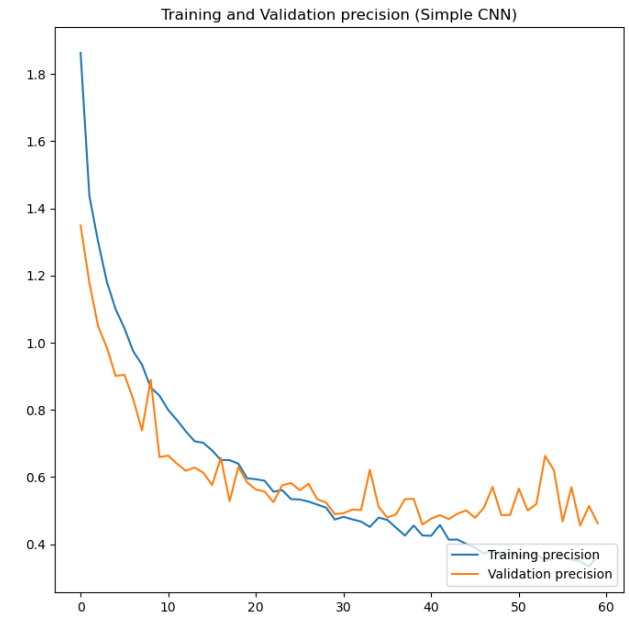
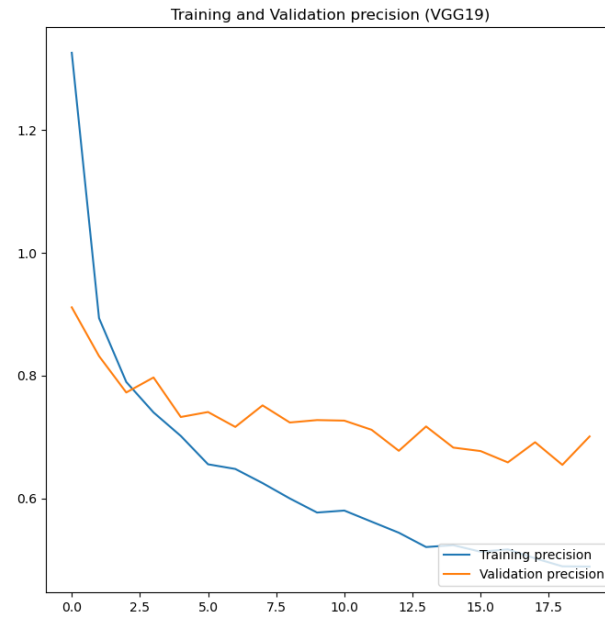
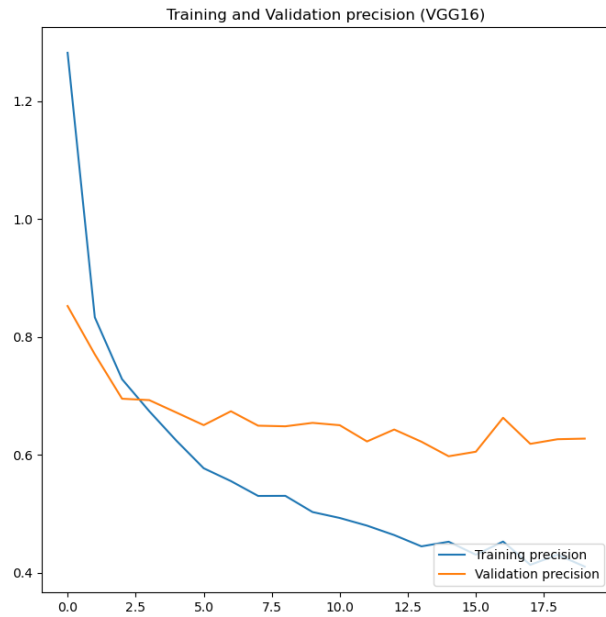
Recall



F1_score



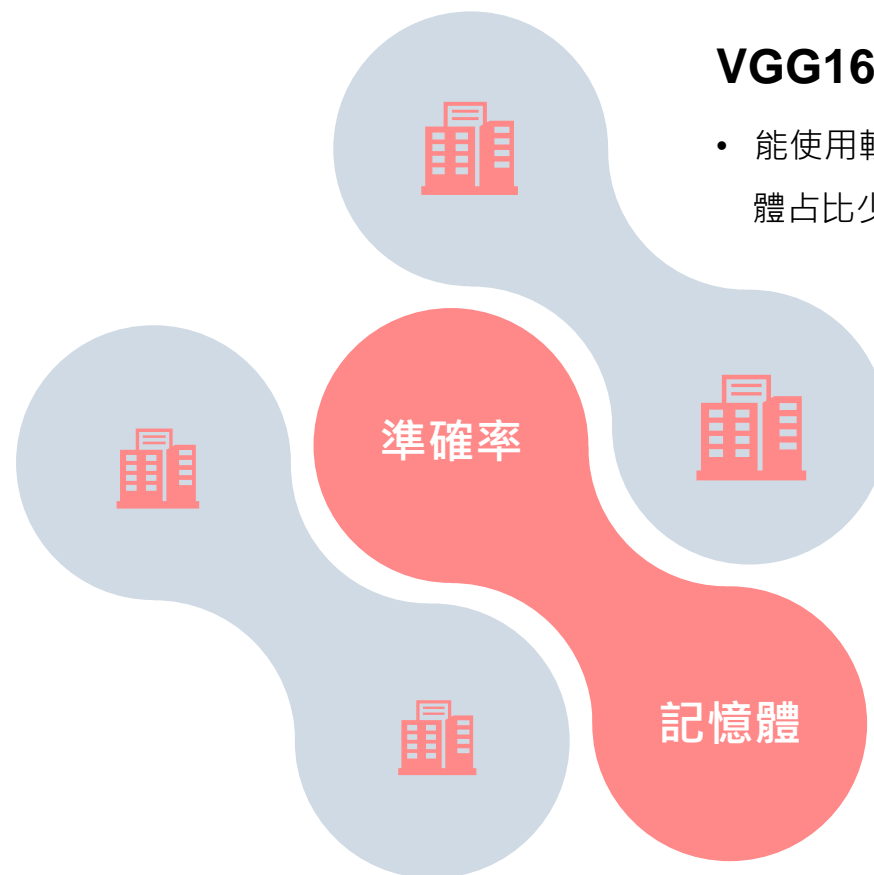
Precision



效能比較

Simple CNN

- SimpleCNN，沒有比以建設好的VGG模型來的有效率，但對學生來說較易理解，也能從中學習各自的差異去改進



VGG16

- 能使用較少epoch數，達到與simpleCNN水準的效果，記憶體占比少

VGG19

- 能使用較少epoch數，達到simpleCNN水準的效果，記憶體占比少

04

The fourth chapter

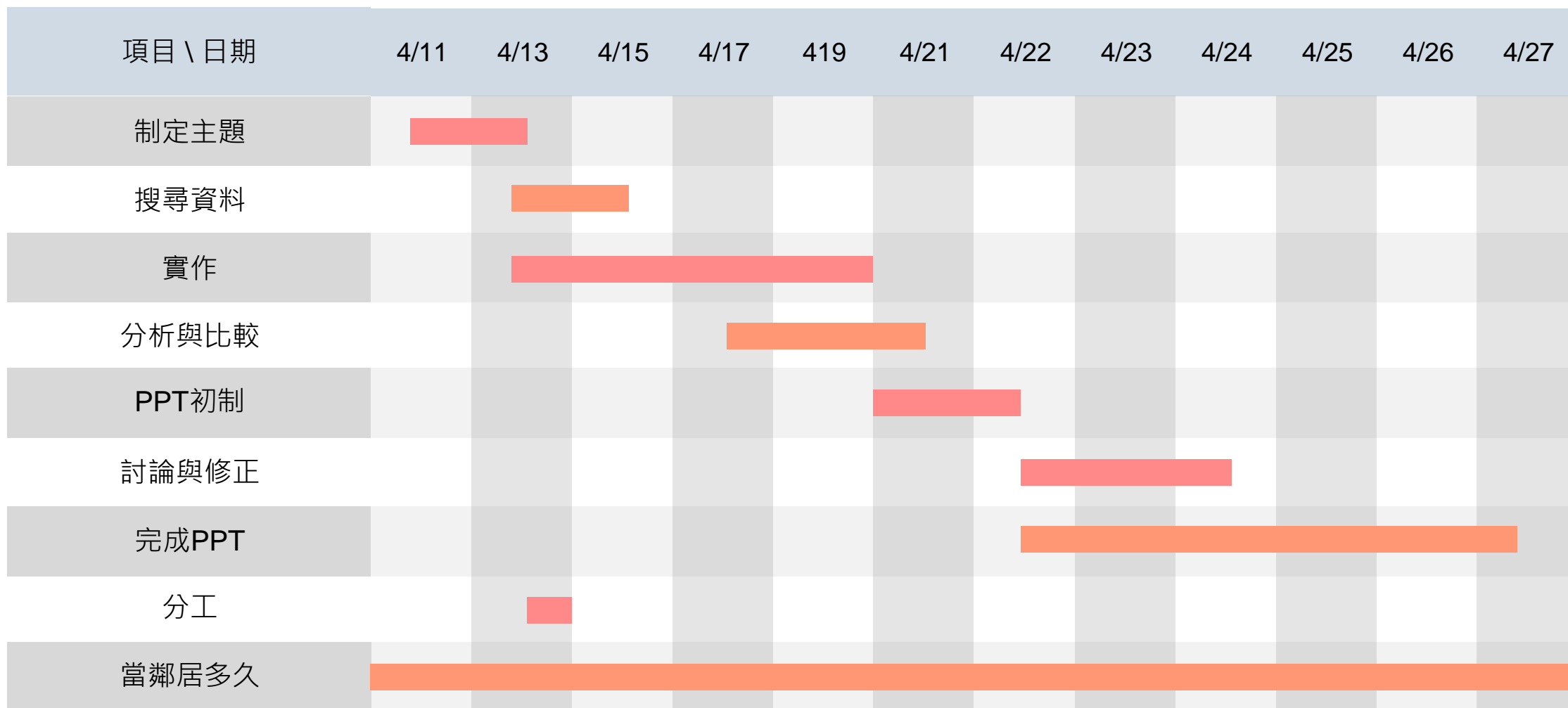
分工及時程表

劉彥汝大發!!!

分工表

	4/11	4/15	4/19	4/23	4/27
楊樸霖 1093210005			PPT製作		
沈奕菱 109321020			PPT製作		
劉彥汝 109321041		蒐集資料及實作			
全員	制定主題	分工	分析與比較	討論及修正	

甘特圖



05

The fifth chapter

遇到的困難

堅持下去就是我的啦~~~哈哈哈哈哈

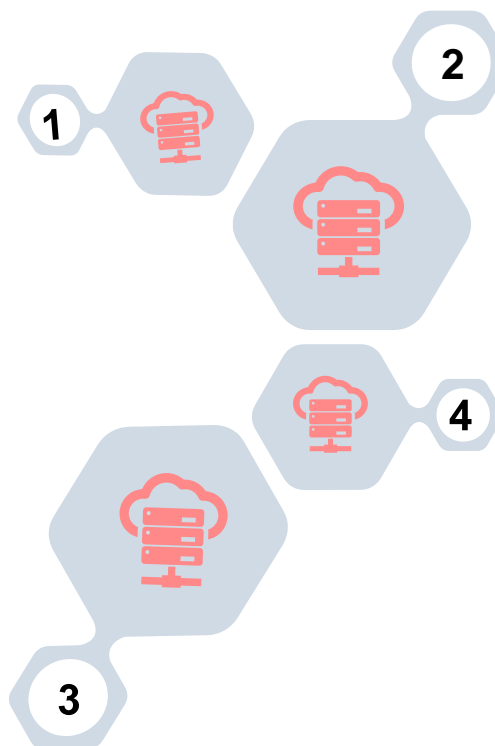
遇到的困難

資料前處理

- 資料量不夠，所以需要套件來縮放或是切割照片等來提升資料量。。

熟悉度

- python用來做資料前處理的套件真的好多，需要花很多時間來熟悉各個套件的使用方法和使用時機。



顯卡效能不足

- 這次訓練所使用的顯卡是Nvidia GTX1660 oc 6G，顯卡的內存明顯不夠，所以在訓練的時候沒辦法跑到100%。

訓練架構

- 這次其實我也有嘗試用EfficientNet，不過效果不像VGG那麼好，所以其實挑對框架比較重要，不是每一種框架都可以用。

06

The sixth chapter

參考資料

讚嘆網路的發達

參考資料



<https://www.kaggle.com>

<https://stackoverflow.com>

機器學習實務(MOOCs)

https://www.youtube.com/playlist?list=PLJV_el3uVTsPy9oCRY30oBPNLCo89yu49

<https://ithelp.ithome.com.tw/articles>





THE END

謝謝觀賞

THANKS