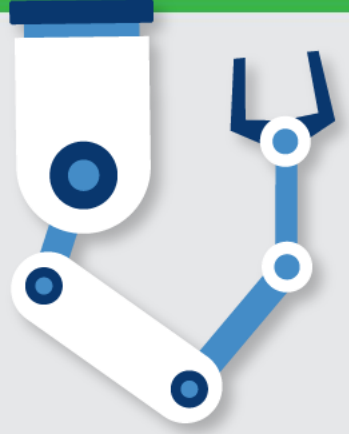
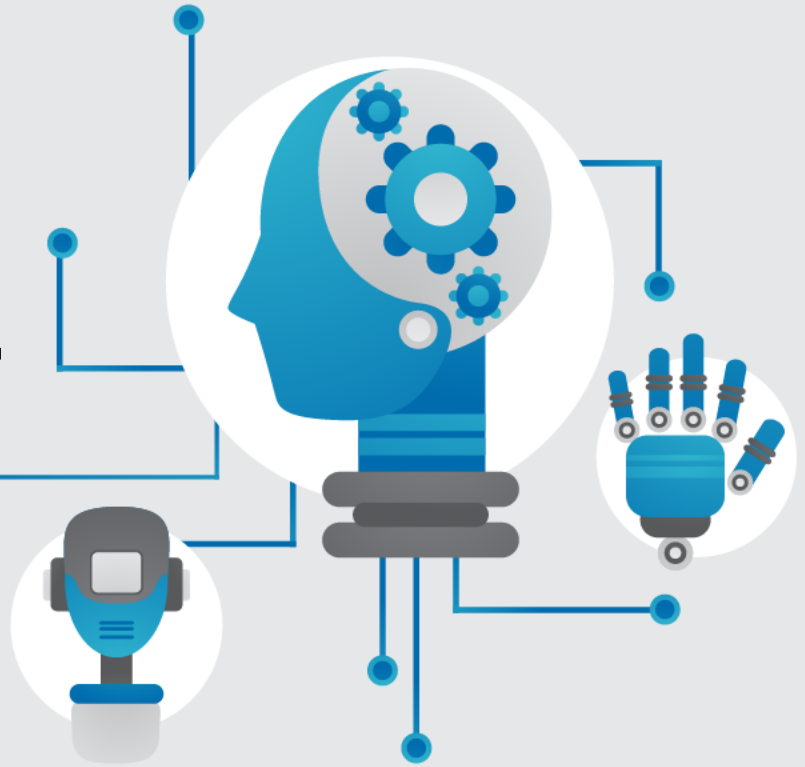


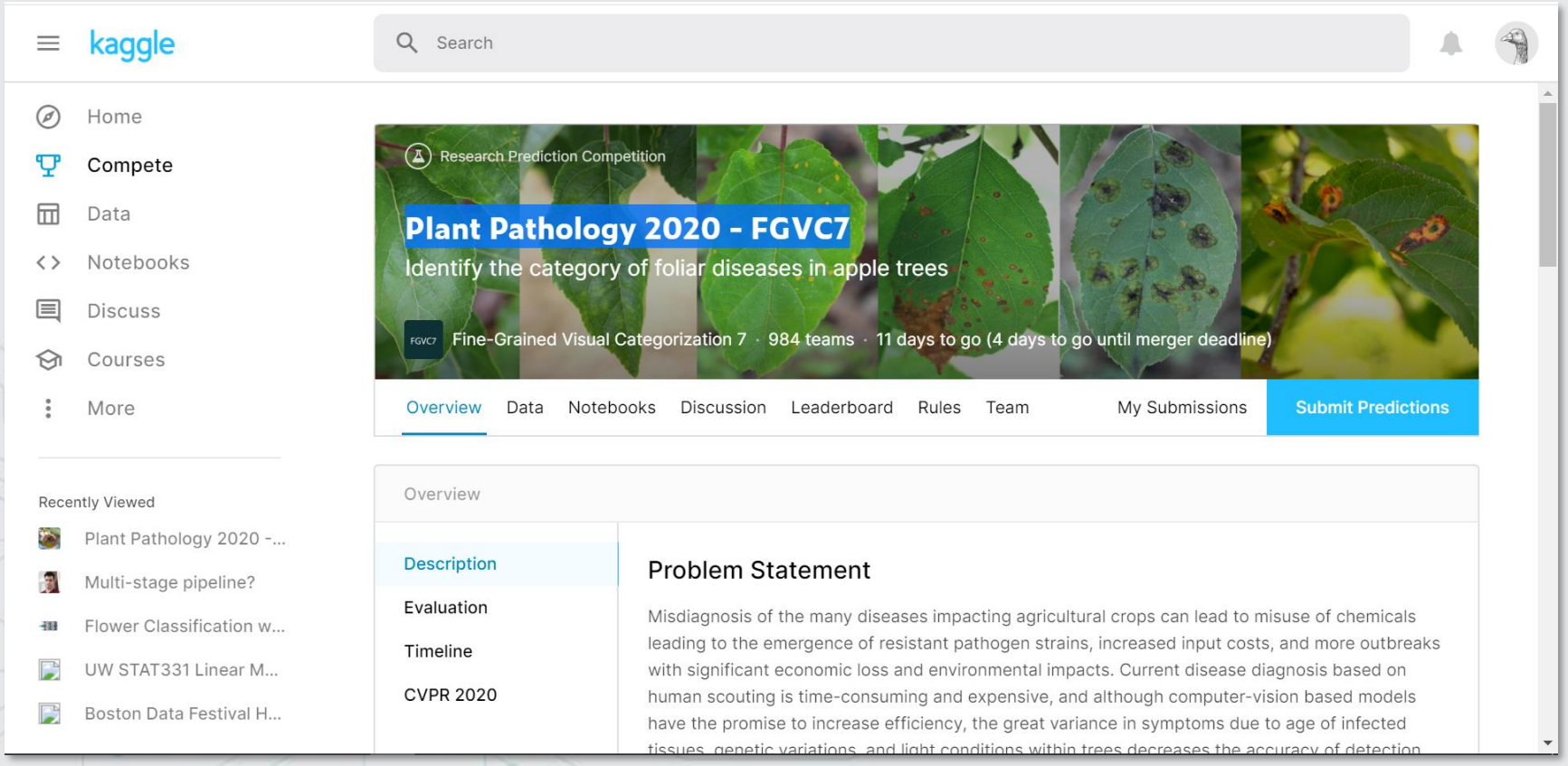
CNN模型訓練與評估



Kaggle競賽 – Plant Pathology 2020 - FGVC7

機器學習實務

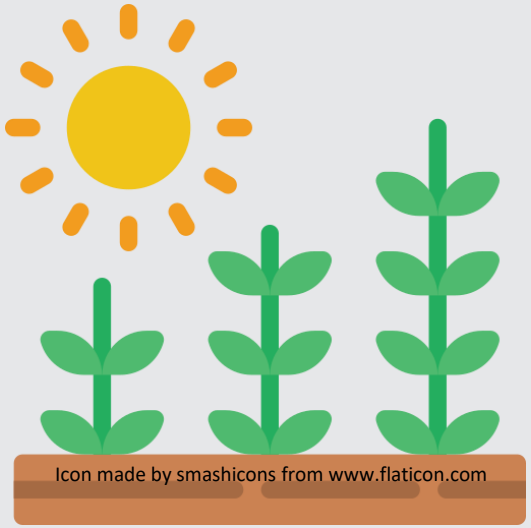
- https://www.kaggle.com/c/plant-pathology-2020-fgvc7



Kaggle競賽 – Plant Pathology 2020 - FGVC7

› 說明

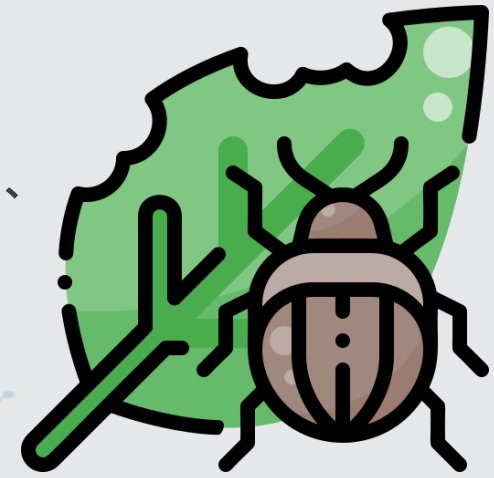
影響農作物的許多疾病的誤診，
會造成化學藥品的濫用，
甚至導致耐藥菌病原體的出現。



當前由人來判定疾病既耗時又昂貴，儘管基於**電腦視覺的模型**有望提高效率，但是由於受感染組織的**年齡**、**遺傳變異**和樹木內的**光照條件**等，導致疾病檢測的準確率差異很大。

Kaggle競賽 – Plant Pathology 2020 - FGVC7

- **目標：** “植物病理學挑戰”的目標是使用訓練數據集的圖像來訓練模型，達到下列目標。
  - ✓ 將測試數據集中的圖像準確分類為不同的患病或健康葉片類別
  - ✓ 有時單葉不止一種疾病，要能準確區分多種疾病
  - ✓ 處理少見的類別和新症狀
  - ✓ 解決深度感知問題，例如葉子的**角度**、**光線**、**陰影和生理年齡**
  - ✓ 導入專家知識，協助搜索相關特徵



Icon made by Pixelmeetup from www.flaticon.com

Kaggle競賽 – Plant Pathology 2020 - FGVC7

數據來源

- 由Zach Guillian所收集
- 數據集為蘋果葉，有健康的葉子、感染了蘋果鏽（apple rust）的葉子、患有蘋果黑星病（apple scab）的葉子和感染多重疾病的葉子。

Icon made by Nhor Phai from www.flaticon.com

Kaggle競賽 – Plant Pathology 2020 - FGVC7

- 資料檔

train.csv

	A	B	C	D	E	F
1	image_id	healthy	multiple_c	rust	scab	
2	Train_0	0	0	0	1	
3	Train_1	0	1	0	0	

test.csv

	A	B
1	image_id	
2	Test_0	
3	Test_1	

sample_submission.csv

	A	B	C	D	E	F
1	image_id	healthy	multiple_c	rust	scab	
2	Test_0	0.25	0.25	0.25	0.25	
3	Test_1	0.25	0.25	0.25	0.25	

Kaggle競賽 – Plant Pathology 2020 - FGVC7

- 葉子照片數量
  - healthy : 516
  - rust : 622
  - scab : 592
  - multiple_diseases : 91

Kaggle競賽 – Plant Pathology 2020 - FGVC7

- **資料問題分析**
  - 照片大小：2048 X 1365
  - 照片數量少，且多重疾病者不平均
  - 照片光線、背景物（樹枝、水果、手）等的干擾

  - 病徵色澤明顯

CNN模型建置流程


1. 資料前處理
2. 決定模型架構與參數
3. 模型編譯和訓練
4. 模型評估
5. 調整超參數
6. 重複步驟2~5
直到模型效率無法再改進
7. 進行預測

資料前處理

- 影像資料擴增（Data augmentation）
  - 旋轉（rotation）
  - 位移（shift）
  - 放大縮小（zoom）
  - 亮度（brightness）

- 使用套件

```
from numpy import expand_dims
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import ImageDataGenerator
```

影像資料擴增

旋轉

```
# rotation augmentation
data = img_to_array(img)
# expand dimension to one sample
samples = expand_dims(data, 0)
# create image data augmentation generator
datagen = ImageDataGenerator(rotation_range=80)
# prepare iterator
it = datagen.flow(samples, batch_size=1)
# generate samples
for i in range(2):
    # generate batch of images
    batch = it.next()
    # convert to unsigned integers for viewing
    image = batch[0].astype('uint8')
```

影像資料擴增

- 旋轉

影像資料擴增

- 位移

```
# horizontal shift augmentation
data = img_to_array(img)
# expand dimension to one sample
samples = expand_dims(data, 0)
# create image data augmentation generator
datagen = ImageDataGenerator(width_shift_range=[-200,200])
# prepare iterator
it = datagen.flow(samples, batch_size=1)
# generate samples
for i in range(2):
    # generate batch of images
    batch = it.next()
    # convert to unsigned integers for viewing
    image = batch[0].astype('uint8')
```

影像資料擴增

- 位移

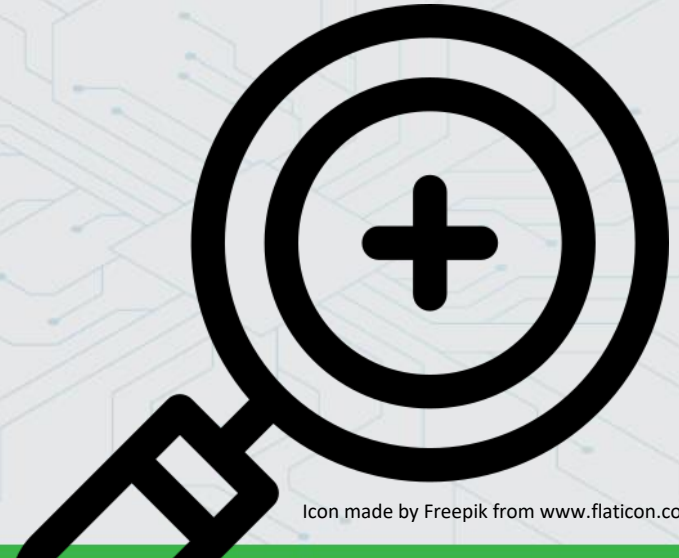
影像資料擴增

- 放大縮小

```
# zoom augmentation
data = img_to_array(img)
# expand dimension to one sample
samples = expand_dims(data, 0)
# create image data augmentation generator
datagen = ImageDataGenerator(zoom_range=[0.5,1.0])
# prepare iterator
it = datagen.flow(samples, batch_size=1)
# generate samples
for i in range(2):
    # generate batch of images
    batch = it.next()
    # convert to unsigned integers for viewing
    image = batch[0].astype('uint8')
```

影像資料擴增

- 放大縮小



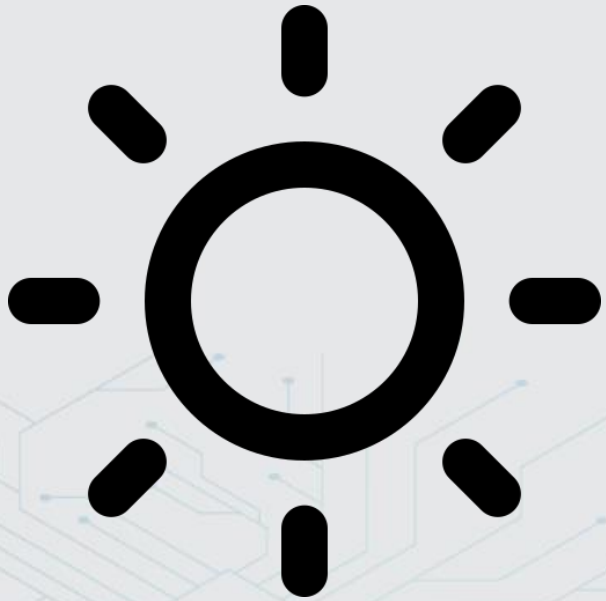
影像資料擴增

亮度

```
# brightness augmentation
data = img_to_array(img)
# expand dimension to one sample
samples = expand_dims(data, 0)
# create image data augmentation generator
datagen = ImageDataGenerator(brightness_range=[0.2,1.0])
# prepare iterator
it = datagen.flow(samples, batch_size=1)
# generate samples
for i in range(2):
    # generate batch of images
    batch = it.next()
    # convert to unsigned integers for viewing
    image = batch[0].astype('uint8')
```

影像資料擴增

- 亮度



模型架構

種類

1. 自訂模型（單一模型，四種類別）+ **圖片不擴增**
2. 自訂模型（單一模型，四種類別）+ **圖片擴增 + class_weight**
3. 自訂模型（三模型：是否healthy？是否有rust ？是否有scab）
   + **圖片不擴增** + class_weight
4. 自訂模型（三模型：是否healthy？是否有rust ？是否有scab）
   + **圖片擴增** + class_weight

訓練結果

- 自訂模型（單一模型，四種類別）+ **圖片不擴增**

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 64, 64, 64)        1792
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 64, 64, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 32, 128)       73856
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 32, 32, 128)       147584
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 128)       0
_________________________________________________________________
```

訓練結果

- 自訂模型（單一模型，四種類別）+ **圖片不擴增**

dropout_2 (Dropout)	(None, 16, 16, 128)	0
conv2d_5 (Conv2D)	(None, 16, 16, 256)	295168
conv2d_6 (Conv2D)	(None, 16, 16, 256)	590080
max_pooling2d_3 (MaxPooling2	(None, 8, 8, 256)	0
dropout_3 (Dropout)	(None, 8, 8, 256)	0
flatten_1 (Flatten)	(None, 16384)	0

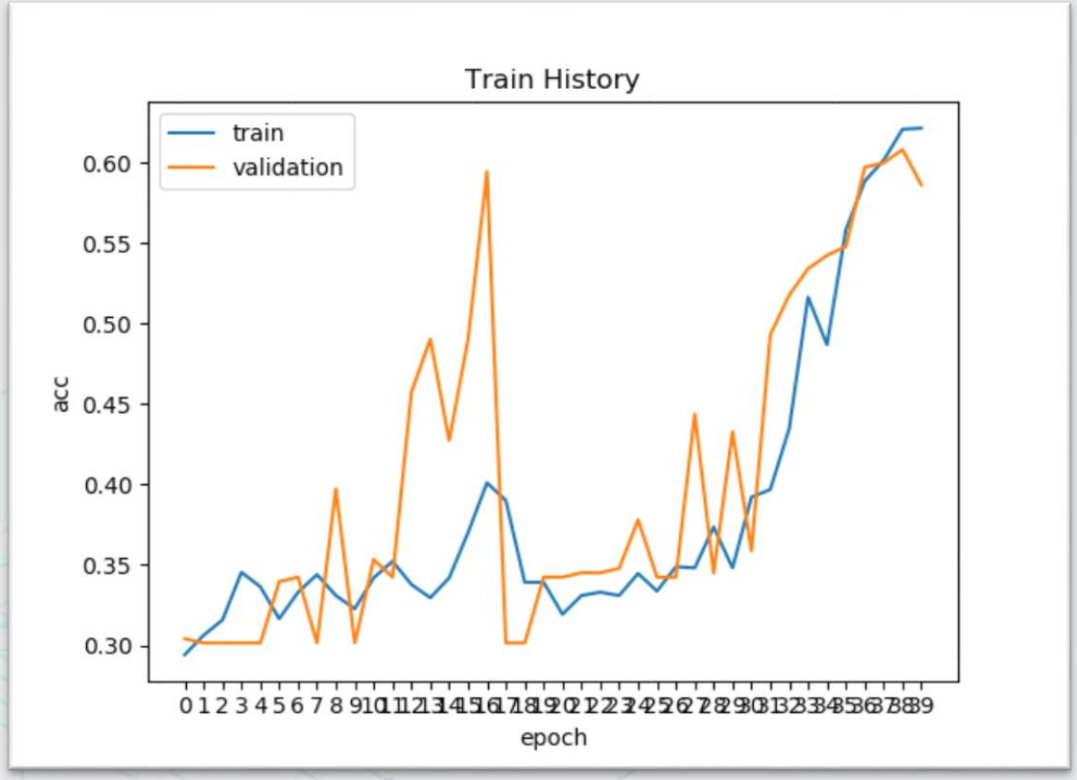
訓練結果

› 自訂模型（單一模型，四種類別）+ 圖片不擴增

dense_1 (Dense)	(None, 128)	2097280
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_5 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
dropout_6 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 4)	132

訓練結果

- 自訂模型（單一模型，四種類別）+ **圖片不擴增**
- 訓練結果

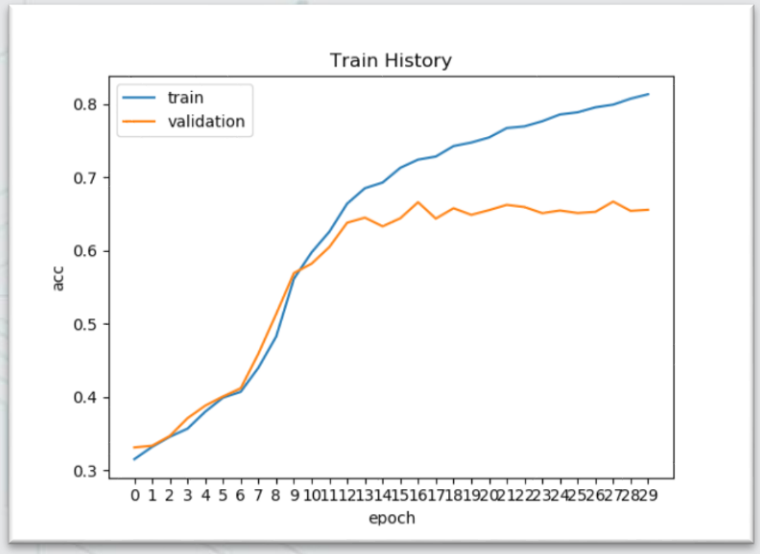


訓練結果

- 自訂模型（單一模型，四種類別）+ 圖片擴增+ class_weight
- 計算class_weight

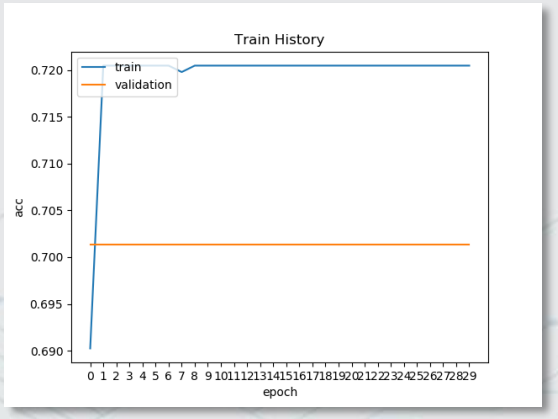
```
from sklearn.utils import class_weight
class_weights =
class_weight.compute_class_weight('balanced',np.unique(y),y)
```

- 訓練結果



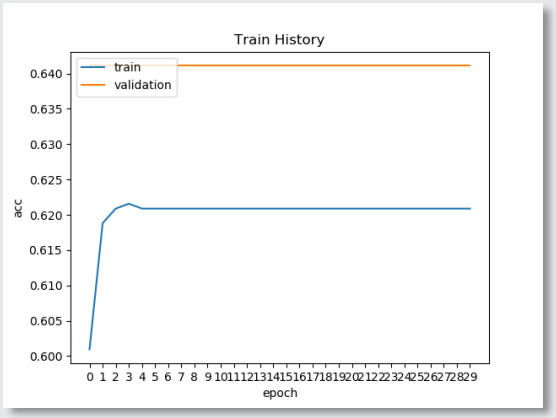
訓練結果

- 自訂模型（三模型：是否healthy？是否有rust？是否有scab）+ **圖片不擴增** + class_weight
- 訓練結果



healthy

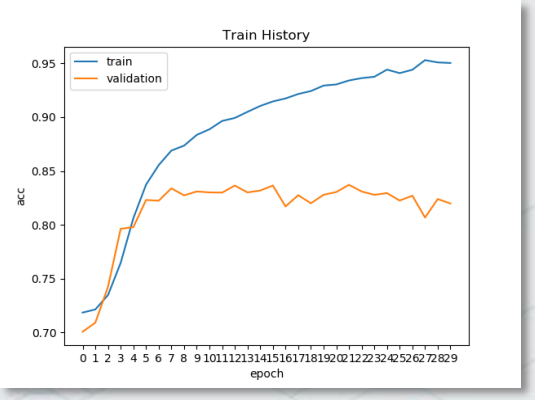
rust



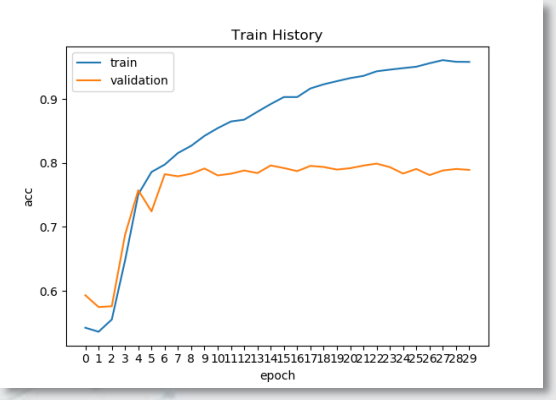
scab

訓練結果

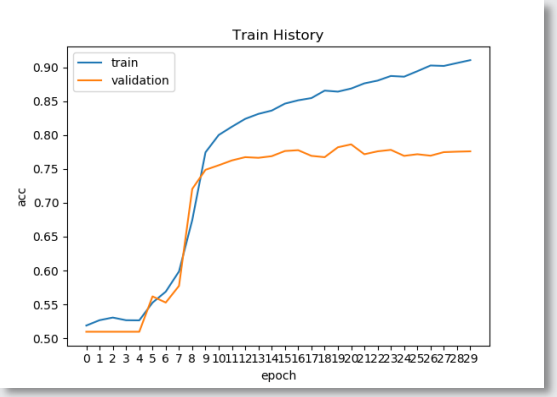
- 自訂模型（三模型：是否healthy？是否有rust？是否有scab）+ **圖片擴增** + class_weight
- 訓練結果



healthy



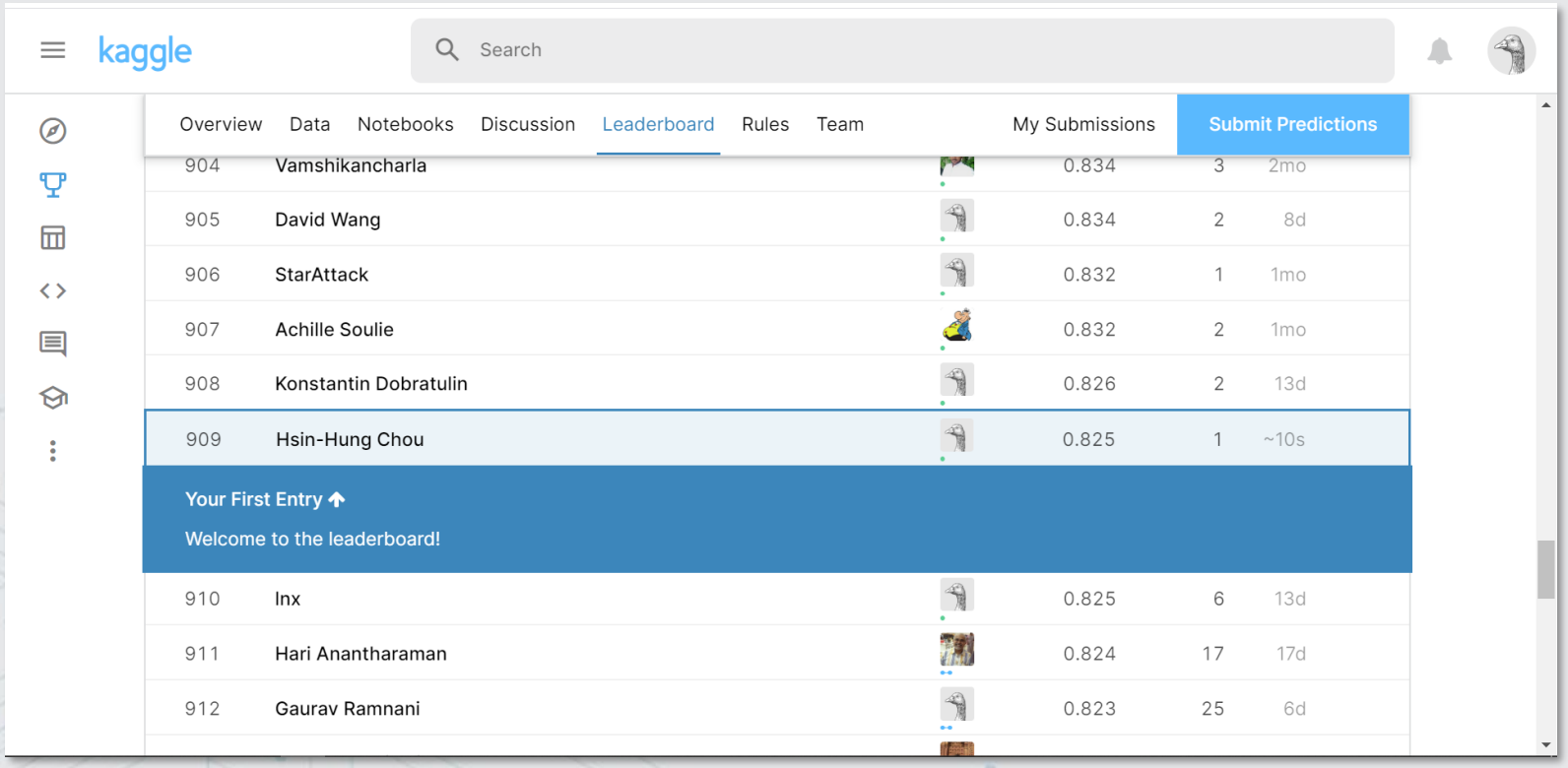
rust



scab

進行預測

- 將預測結果上傳Kaggle平台



討論

訓練資料少

- Data augmentation
  - ✓影像處理（旋轉、位移、亮度等），效果有限
  - ✓生成對抗網路（GAN）
- Segmentation （U-net, R-CNN 等）

資料不平均

- 使用類別權重（class weight）
- 擴增或減少資料