

【附件 1】**臺北市立中正高級中學 111 學年度校內科學展覽
報名表暨內容說明表**

作品名稱		使用 AI 圖像辨識進行垃圾分類之探討		
科 別		電腦與資訊學科		
作 者	班級	310	310	
	座號	08	18	
	姓名	朱容徵	林晉霆	
作品內容說明簡介(100 字以上) 由於觀察到社會大眾對於人行道上垃圾桶進行垃圾分類的不確實分類，於是便想藉由 AI 工具來幫助民眾瞭解手上的垃圾應該分類到的正確類別，以至於達到垃圾分類正確率上升的研究目的。 本次科展會藉由不同種類的 AI（監督式學習的：決策樹、梯度提升樹、隨機森林、貝氏分類器、邏輯回歸、支持向量機、捲積神經網路、YOLO；非監督式學習的：K-means）分類模型加上圖像辨識功能，以達到辨識、分類不同種類的垃圾的研究方法，最後以 AI 模型為實驗組，人類作為對照組，在固定時間內對相同垃圾進行垃圾分類，最後對其做比對和研究結論。				
學科指導教師		賴敏芳 老師	導師/社團指導師	陳華傑 老師

◎ 報名注意事項：

1. 111 年 1 月 11 日（星期三）16:00 前繳交「科展報名表暨內容說明表」完成報名手續。

【附件 2】

**臺北市立中正高中 111 學年度科學展覽會
作品說明書**

科 別：電腦與資訊學科

組 別：高級中等學校組

作品名稱：使用 AI 圖像辨識進行垃圾分類之探討

關 鍵 詞： AI 、 圖像辨識 、 垃圾分類 （最多 3 個）

編 號：

製作說明：

- 1.說明書封面僅寫科別、組別、作品名稱及關鍵詞。
- 2.編號由設備組統一編列。
- 3.封面編排由參展作者自行設計。

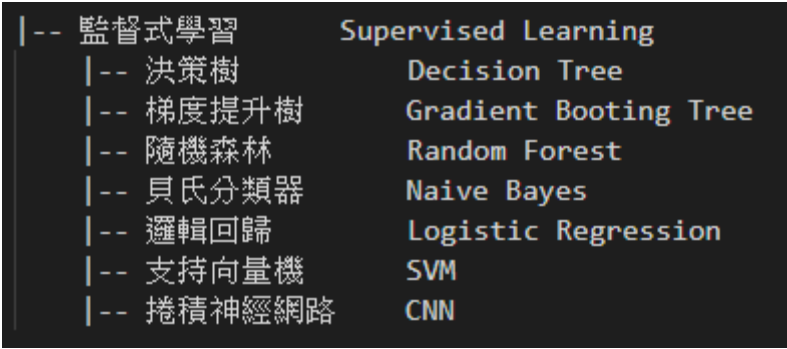
臺北市立中正高中 111 學年度科學展覽會作品說明書內容說明

作品名稱：使用 AI 圖像辨識進行垃圾分類之探討

摘要

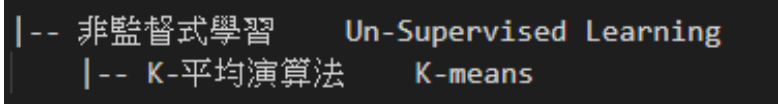
本次科展會使用不同種類的 AI（監督式學習的：決策樹、梯度提升樹、隨機森林、貝氏分類器、邏輯回歸、支持向量機、捲積神經網路、YOLO；非監督式學習的：K-means）實作圖像辨識 AI。

將以各種 AI 模型作為實驗組、真人作為對照組，在固定時間內對相同數量、種類垃圾進行垃圾分類，以對比不同 AI 在垃圾分類上的正確率以尋找最適合用於進行垃圾分類的圖像辨識 AI。



-- 監督式學習	Supervised Learning
-- 決策樹	Decision Tree
-- 梯度提升樹	Gradient Boosting Tree
-- 隨機森林	Random Forest
-- 貝氏分類器	Naive Bayes
-- 邏輯回歸	Logistic Regression
-- 支持向量機	SVM
-- 捲積神經網路	CNN

（圖一，監督式學習模型種類）



-- 非監督式學習	Un-Supervised Learning
-- K-平均演算法	K-means

（圖二，非監督式學習模型種類）

壹、研究動機

最近發現很多人行道上環保局設立的垃圾桶被許多未確實分類的垃圾堆滿，而近看可以發現其中資源回收類跟一般垃圾類都混雜了許多放置錯誤的垃圾以及家用垃圾，例如：吸管應該丟入一般垃圾類，而不是資源回收。如此錯誤的垃圾處理方式只會造成更嚴重的空汙問題以及資源回收物的總盈餘降低，成為社會上的惡性循環、負面成本。

由於垃圾分類的亂象發生，因此我們希望利用所學到的 AI 知識設計一個 AI 圖像辨識模型，可以通過攝影機對垃圾物件作出分類，希望能改善此情況。

貳、研究目的

一、可以成功訓練出具有高正確率分辨垃圾分類的 AI 模型

能夠成功訓練出任何一種對於垃圾分類有高正確率(準確率在 90%以上)的 AI 模型。

二、探究不同 AI 模型在垃圾分類上的正確率及其原因

以上述各種 AI 模型(圖一、圖二)訓練其可以對垃圾進行分類後，以不同學習模型在對未標記特徵垃圾進行分類時，所最後統計出的正確率做數據分析，並對其數據呈現(例如：正確率最高者、最低者)做研究探討。

三、可以在垃圾分類的對照組與實驗組上面得到 AI 對其的影響

回歸科展初衷，目的是希望能夠利用 AI 這項工具改善大眾對於垃圾分類這項作為的準確度提高，以 AI 降低大眾亂丟垃圾和垃圾分錯類的現象。

參、研究設備及器材

一、硬體設備

(一)筆記型電腦

筆電型號	Lenovo Y520
作業系統	Windows 10 Home
CPU	Intel Core i5-7300HQ 2.50 GHz
GPU	NVIDIA GeForce GTX 1050 Ti (notebook)
VRAM	4 GB
RAM	16 GB

(二)攝影鏡頭

品牌	解析度
EDSDS 愛迪生 網路攝影機	480P

二、軟體設備

(一)軟體：

VScode、Visual Studio Code 2017 Community、CUDA 10.0、cuDNN 7.4

(二)模組(Module)：

Tensorflow 2.0.0、Keras 2.4.7、numpy、matplotlib、opencv、pillow

(三)工作平台：

Google Colab

肆、研究過程或方法

本次 AI 模型著重在於圖像處理（image preprocessing）跟圖像分類（image classification），將透過不同 AI 分類模型對垃圾分類的準確率，找出對垃圾分類準確度最高的模型，最後將其作為解決垃圾亂丟問題的解決方法。

具體步驟流程如下：

一、資料前處理

(一)蒐集所有關於垃圾的圖片檔¹，並統一轉成.jpg 檔，並將每一種垃圾種類做好個別的資料夾（圖三）

(二)將不同垃圾種類正確分成兩個類別：資源回收（recyclable）、一般垃圾（not recyclable）

(三)製作訓練集（train_data）、測試集（test_data）資料夾，並對該目錄下.jpg 資料夾圖片數量進行分割（圖四）²

```
|-- straw                                |-- Trash_Classification
|   |-- jpg                             |-- test
|   |   |-- (1).jpg                     |-- not_recyclable
|   |   |-- (2).jpg                     |-- ...
|   |   |-- (...).jpg                  |-- recyclable
|   |-- txt                             |-- jpg
|   |   |-- (1).txt                     |-- ...
|   |   |-- (2).txt                     |-- txt
|   |   |-- (...).txt                  |-- ...
|   |-- 吸管.txt                       |-- (該垃圾樣式名稱).txt
|                                     |-- train
|                                     |-- ...
```

（圖三，個別垃圾種類資料目錄分布、圖四，垃圾分類資料夾整體資料目錄分布）

¹ [垃圾種類蒐集品種及蒐集張數](#)

² 訓練集照片數量：測試集照片數量 = 0.9 : 0.1

二、資料載入並轉換成訓練資料

因為資料獲取的不足，我們將採用 keras 內的 ImageDataGenerator() 進行資料增強³的動作，但是因為此方法轉移出來的資料不能直接給 DecisionTreeClassifier() 等模型訓練（CNN，Sequential() 可以），他會是如圖八的資料型態，而通常進行模型訓練需要載入的資料型態為 numpy.ndarray（array 的形式），於是需要進行圖九的動作。

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

（圖五，導入 ImageDataGenerator()）

```
[6] train_data = ImageDataGenerator(rescale = 1.0 / 255)
    test_data = ImageDataGenerator(rescale = 1.0 / 255)
```

（圖六，對變數 train_data、test_data 做定義）

```
[7] train_data = train_data.flow_from_directory(
    directory = TRAIN_PATH,
    batch_size = BATCH_SIZE,
    target_size = (IMG_SIZE, IMG_SIZE),
    class_mode = "categorical")

test_data = test_data.flow_from_directory(
    directory = TEST_PATH,
    batch_size = BATCH_SIZE,
    target_size = (IMG_SIZE, IMG_SIZE),
    class_mode = "categorical")

Found 237 images belonging to 2 classes.
Found 28 images belonging to 2 classes.
```

（圖七，對 train_data、test_data 載入圖片和定義資料）

```
[8] print(type(train_data))

<class 'keras.preprocessing.image.DirectoryIterator'>
```

（圖八，查看 train_data 資料型態）

³ 利用現有的資料經過旋轉、翻轉、縮放…等等方式增加更多的訓練資料。


```
[62] train_x = np.concatenate([train_data.next()[0] for i in range(train_data.__len__())])
    train_y = np.concatenate([train_data.next()[1] for i in range(train_data.__len__())])

    test_x = np.concatenate([test_data.next()[0] for i in range(test_data.__len__())])
    test_y = np.concatenate([test_data.next()[1] for i in range(test_data.__len__())])
```

(圖九對 keras.preprocessing.image.DirectoryIterator 進行轉換)

```
[63] print(type(train_x))
    print(train_x.shape)
    print(train_y.shape)

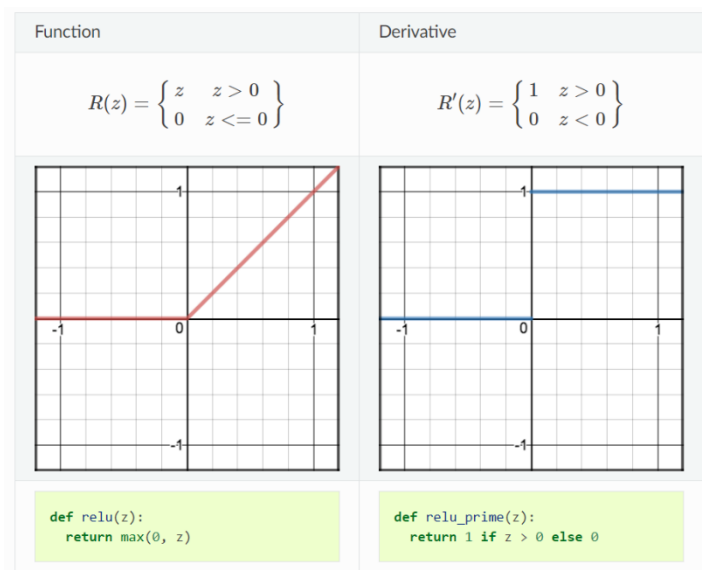
<class 'numpy.ndarray'>
(237, 256, 256, 3)
(237, 2)
```

(圖十⁴，查看 train_data、test_data 資料型態)

⁴ train_x，train_y 是 train_data 拆開、轉型而已，同理 test_data

三、使用並訓練 tensorflow、keras 模型

使用 google colab 進行線上模型訓練，基於 tensorflow 模組去訓練模型，將以 CNN 作為操作範例。CNN 激發函數將使用 ReLu，因為其計算快速和 all or none law 特性，不會產生神經衝動（圖十、十三）。再者，因為是分成資源回收、一般垃圾，因此損失函數採用 binary_crossentropy()（圖十一），再加優化器（optimizer，使用 Adam）去優化模型訓練結果，並避免模型學習更新太慢或無法收斂，因此限定學習率（圖十二）。



$$R(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}$$

（圖十，Relu，函式特性及數學式）

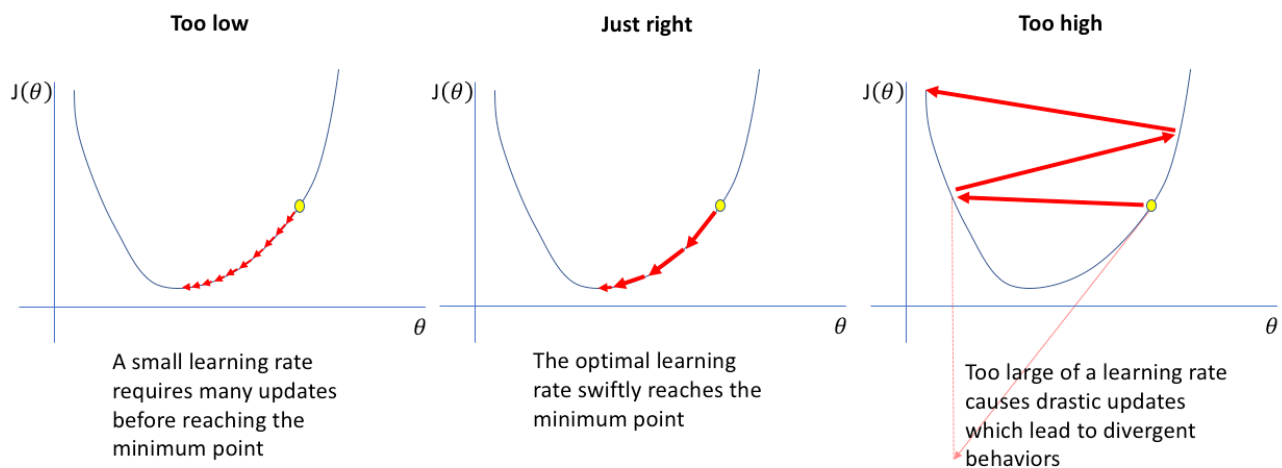
$$\text{loss} = - \sum_{i=1}^n \hat{y}_i \log y_i + (1 - \hat{y}_i) \log(1 - \hat{y}_i)$$

$$\frac{\partial \text{loss}}{\partial y} = - \sum_{i=1}^n \frac{\hat{y}_i}{y_i} - \frac{1 - \hat{y}_i}{1 - y_i}$$

5

（圖十一，binary_crossentropy 交叉熵損失函數，機率相差越大，Loss 就越大）

⁵ 只有 y_i 和 \hat{y}_i 是相等時，loss 才為 0，否則 loss 就是為一個正數



(圖十二，學習率大小和其模型訓練關係)

```
[9] model = Sequential()

[10] model.add(Conv2D(16, kernel_size = (3, 3), activation = "relu", input_shape = (IMG_SIZE, IMG_SIZE, 3)))

    model.add(Conv2D(32, kernel_size = (3, 3), activation = "relu"))
    model.add(MaxPooling2D(pool_size = (2, 2)))

    model.add(Dropout(0.25))

    model.add(Conv2D(64, kernel_size = (3, 3), activation = "relu"))
    model.add(MaxPooling2D(pool_size = (2, 2)))

    model.add(Conv2D(128, kernel_size = (3, 3), activation = "relu"))
    model.add(MaxPooling2D(pool_size = (2, 2)))

    model.add(Conv2D(128, kernel_size = (3, 3), activation = "relu"))
    model.add(MaxPooling2D(pool_size = (2, 2)))

    model.add(Dropout(0.25))

    model.add(Flatten())

    model.add(Dense(64, activation = "relu"))

    model.add(Dropout(0.5))

    model.add(Dense(CLASSES, activation = "sigmoid"))
```

(圖十三，CNN 模型架構圖)

```
[11] model.compile(loss = "binary_crossentropy", optimizer = Adam(learning_rate = 0.0001, decay = 1e-6), metrics = ["accuracy"])
```

(圖十四，CNN 組裝)

```
[13] model_info = model.fit(train_data, epochs = EPOCH, validation_data = test_data)
```

(圖十五，CNN 訓練過程)

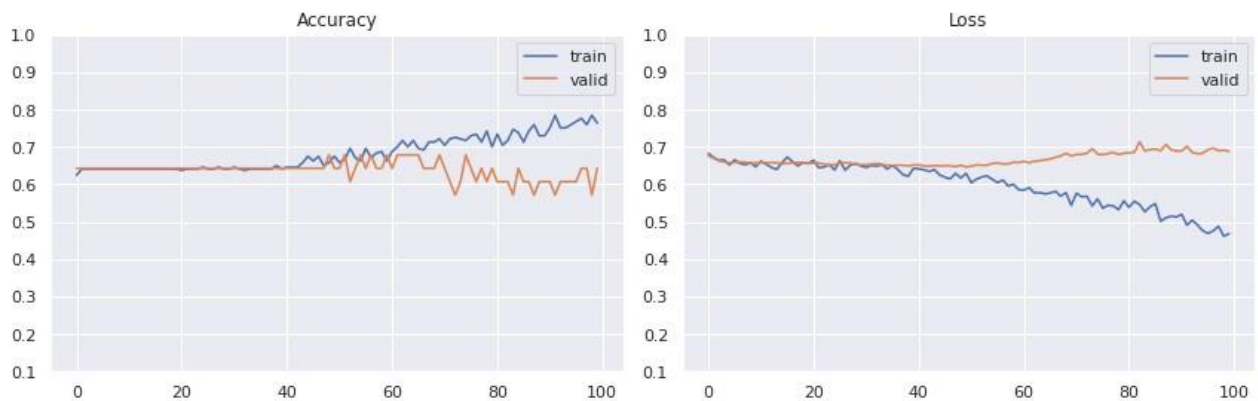
```
Epoch 95/100
8/8 [=====] - 13s 2s/step - loss: 0.4771 - accuracy: 0.8101 - val_loss: 0.6146 - val_accuracy: 0.5714
Epoch 96/100
8/8 [=====] - 13s 2s/step - loss: 0.4959 - accuracy: 0.7468 - val_loss: 0.6408 - val_accuracy: 0.6429
Epoch 97/100
8/8 [=====] - 13s 2s/step - loss: 0.4917 - accuracy: 0.7848 - val_loss: 0.6219 - val_accuracy: 0.5357
Epoch 98/100
8/8 [=====] - 13s 2s/step - loss: 0.4725 - accuracy: 0.7932 - val_loss: 0.6375 - val_accuracy: 0.6071
Epoch 99/100
8/8 [=====] - 12s 2s/step - loss: 0.4825 - accuracy: 0.7764 - val_loss: 0.6525 - val_accuracy: 0.6429
Epoch 100/100
8/8 [=====] - 12s 2s/step - loss: 0.4768 - accuracy: 0.7806 - val_loss: 0.6296 - val_accuracy: 0.5357
```

(圖十六，CNN 訓練結果，accuracy 0.7806 為最後訓練結果)

四、輸出資料視覺化

```
import seaborn as sns
import matplotlib.pyplot as plt
```

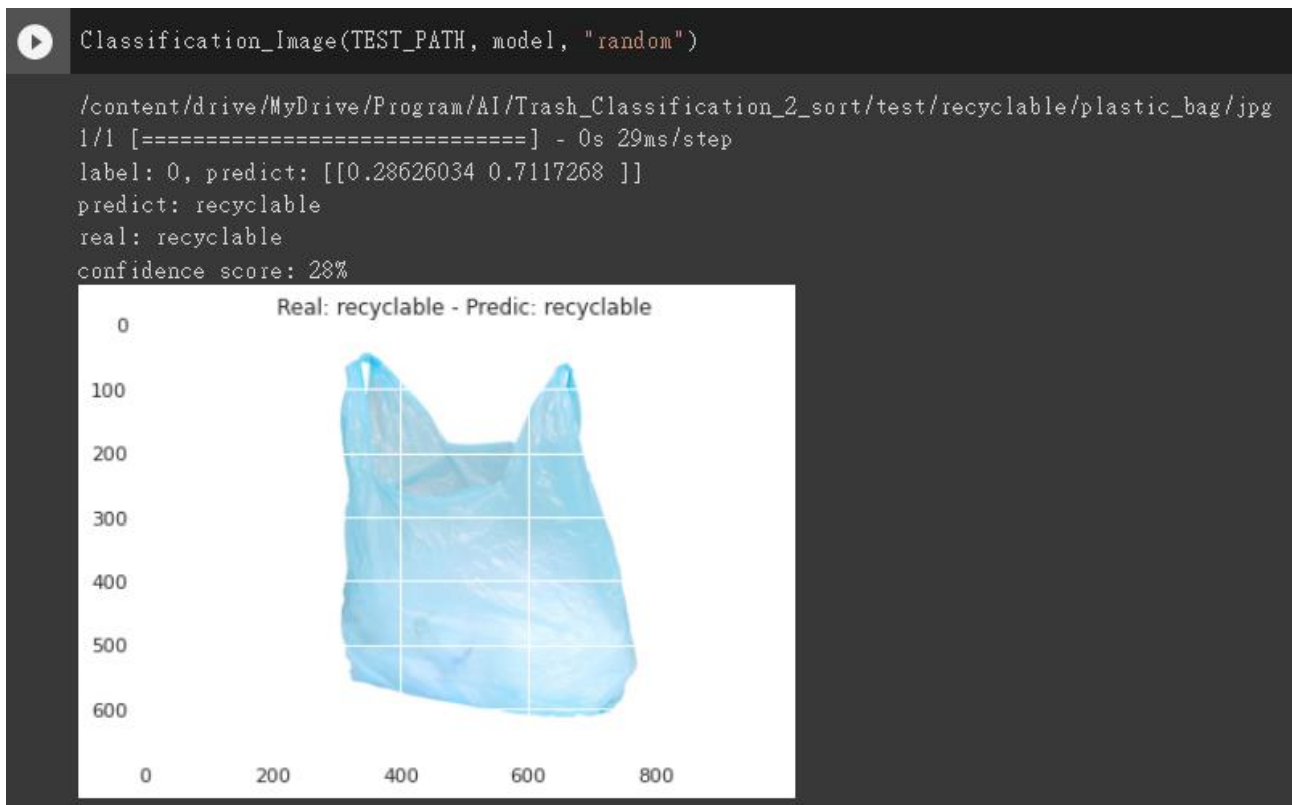
(圖十七，載入視覺化工具)



(圖十八，CNN 訓練過程視覺化)

使用 matplotlib、seaborn 模組繪製模型訓練時狀態 (Accuracy、Loss) 的圖表，和在輸入圖片給模組進行分類時，輸出預測後的圖表內容，可以更直覺的呈現分類結果，如圖十八。

五、輸出模型準確值、分類結果



(圖十九，CNN 輸入圖片進行分類)

轉寫一個叫 Classification_Image()的函式，執行丟入圖片顯示分類結果的功能。放入測試資料後輸出以下資訊：

(一)圖片路徑

(二)label = 0，資源回收；label = 1，一般垃圾

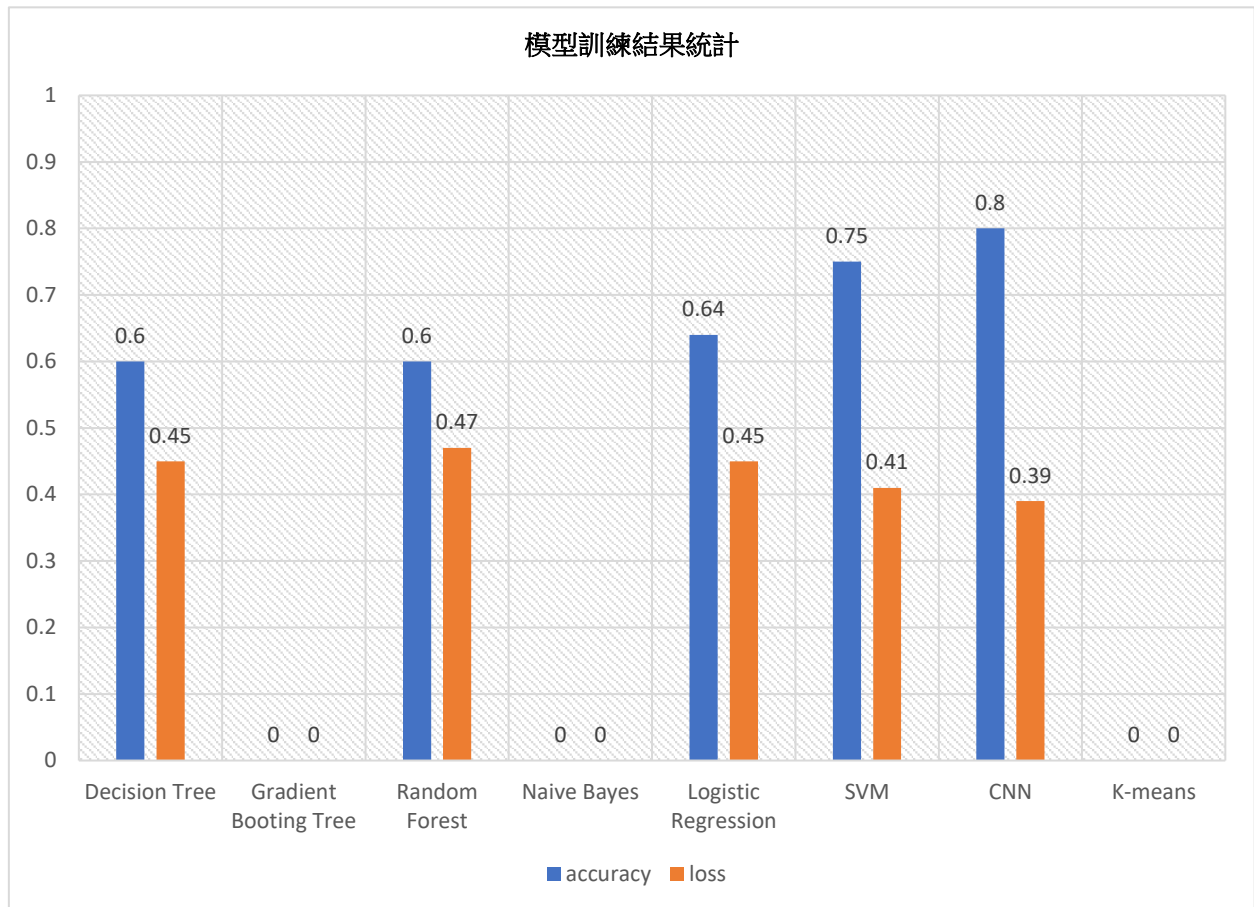
(三)預測類別：資源回收

(四)實際類別：資源回收

(五)信心值：28%

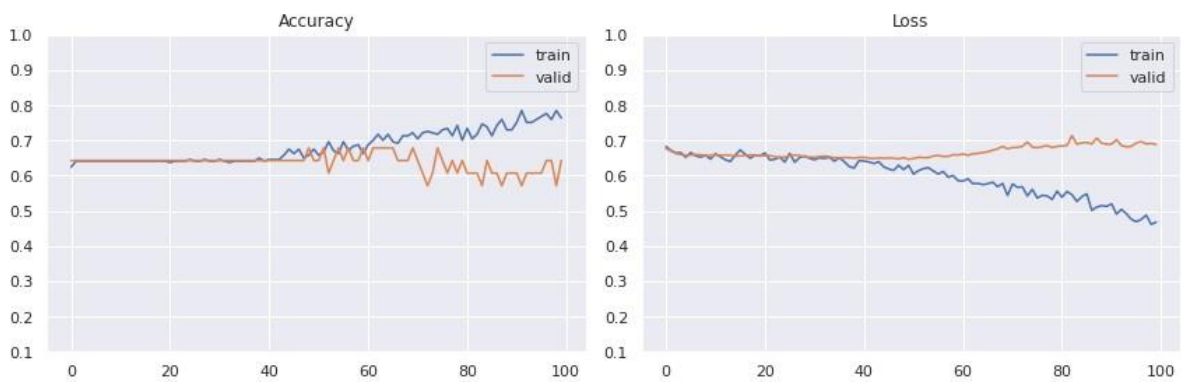
伍、研究結果

一、圖一，圖二模型訓練結果統計



(圖二十，模型準確度量化分析，取對高一次做呈現)

二、以 CNN 作為結果呈現



(一)訓練集準確率：76.37%

(二)驗證集準確率: 64.29%

三、小結

從圖二十一可以看到所有模型的捲積神經網路在圖像辨識上的正確率最高，因此捲積神經網路最適合用來作為垃圾分類 AI 的模型。

陸、討論

一、蒐集的訓練資料可能過少，每一種垃圾的資料都大約是 50 筆，雖然已經有使用 tensorflow 的 ImageDataGenerator 進行資料強化，但搜集的資料量可能還是會影響結果。

二、由於使用的 CNN 模型沒有接受 label 而是使用整張圖片訓練，可能使 AI 學習到一些不相關的特徵從而降低準確度。因此自行使用程式將圖片根據 label 進行切割，並將訓練集換成切割後的資料以提升準確度。

柒、結論

綜合訓練過程和參考圖二十做整體評估，在圖一、二模型中 CNN 為最佳圖像分類模型，如果將 CNN 作為垃圾分類的工具可以達到高效率的成效（訓練時間短、訓練過程錯誤少、模型預測分類簡單，等特點）。

捌、參考資料及其他

一、工作分配

	31008 朱容徽	31018 林晉鑫
word 文書報告		
報名表暨內容說明表	<input checked="" type="checkbox"/>	<input type="checkbox"/>
摘要	<input checked="" type="checkbox"/>	<input type="checkbox"/>
壹、研究動機	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
貳、研究目的	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
參、研究設備及器材	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
肆、研究過程或方法	<input checked="" type="checkbox"/>	<input type="checkbox"/>
伍、研究結果	<input type="checkbox"/>	<input checked="" type="checkbox"/>
陸、討論	<input type="checkbox"/>	<input checked="" type="checkbox"/>
柒、結論	<input checked="" type="checkbox"/>	<input type="checkbox"/>
捌、參考資料及其他	<input checked="" type="checkbox"/>	<input type="checkbox"/>
程式製作		
Decision Tree	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Gradient Booting Tree	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Random Forest	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Naïve Bayes	<input type="checkbox"/>	<input type="checkbox"/>
Logistic Regression	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SVM	<input type="checkbox"/>	<input type="checkbox"/>
CNN	<input checked="" type="checkbox"/>	<input type="checkbox"/>
K-means	<input type="checkbox"/>	<input type="checkbox"/>
資料蒐集	<input type="checkbox"/>	<input checked="" type="checkbox"/>
資料處理	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

6

[DECISION TREE AS AN IMAGE CLASSIFICATION TECHNIQUE](#)

[基於影像辨識之垃圾分類機器人](#)

⁶ [工作分配](#)

書寫說明：

- 1.作品說明書一律以 A4 大小紙張由左至右打字印刷，並裝訂成冊。
- 2.作品說明書內容總頁數以 30 頁為限(不含封面、封底及目錄)。
- 3.內容使用標題次序為壹、一、（一）、1、（1），詳見作品說明書電腦檔案製作規範。
- 4.研究動機內容應包括作品與教材相關性（教學單元）之說明。
- 5.原始紀錄資料須攜往評審會場供評審委員查閱，但請勿將研究日誌或實驗觀察原始紀錄正本或影本送交承辦學校。
- 6.作品說明書自本頁起請勿出現校名、作者、校長及指導教師姓名等，並且照片中不得出現作者或指導教師之臉部，俾審查之公平性及客觀性。
- 7.參考資料書寫方式請參考 APA 格式第六版。

作品說明書電腦檔案製作規範

壹、封面：

- 一、版面設定：上、下、左、右各 2cm
- 二、封面字型：16 級

貳、內頁：

- 一、版面設定：上、下、左、右各 2cm
- 二、字型：新細明體
- 三、行距：1.5 倍行高
- 四、主題字級：16 級粗體、置中
- 五、內文字級：12 級
- 六、項目符號順序：

例：

壹、 XXXXXXXX

一、 XXXXXXXX

(一) XXXXXXXX

1. XXXXXXXX

(1) XXXXXXXX

貳、 OOOOOOOO

一、 OOOOOOOO

(一) XXXXXXXX

1. OOOOOOOO

(1) OOOOOOOO

參、對齊點：使用定位點對齊或表格對齊

一、定位點

AAAAAAA

BBBBBBBBB

CCCCCCC

DDDDDDDD

二、表格

AAAAAA

BBBBBBB

CCCCCCC

DDDDDDDD

肆、電子檔：

- 一、文字與圖表及封面須排版完成於1個檔案中。
- 二、以WORD文件檔（*DOC或*DOCX）及PDF圖檔為限。
- 三、檔案名稱為作品名稱。
- 四、檔案大小限10M Bytes以內。
- 五、一律以內文第一頁起始插入頁碼。