

2025 年【科學探究競賽-這樣教我就懂】

□國中組 普高組 □技高組 成果報告格式

題目名稱：自動分類垃圾桶—深度神經網路結合視覺辨識

一、摘要

本計畫旨在運用深度神經網路 (DNN) 結合視覺辨識技術，提升垃圾分類的準確性與效率。根據聯合國統計全世界每年都會產生超過 20 億噸的垃圾，而其中約 4 億噸的塑膠垃圾卻只有百分之十被回收。然而隨著全球垃圾問題日益嚴重，垃圾分類的方法卻沒有得到妥善的改善。傳統的垃圾分類方式常依賴人工進行，既耗時又容易出錯，因此極需發展自動化技術來解決這一挑戰。深度神經網路在影像辨識領域表現出色，通過對大量垃圾圖像的學習，DNN 可以準確地識別不同類型的垃圾，並自動進行分類。本計畫將建立一個基於深度學習的視覺辨識系統，該系統能夠快速、準確地分析並分類各類垃圾，從而大幅減少人工分類的負擔。

使用深度神經網路進行訓練，使系統能夠識別各種物品 (如塑料、金屬、紙張等)，並根據其特徵進行正確分類。這樣的技術不僅提升了分類效率，還能降低分類錯誤率。核心目標是利用先進的視覺辨識技術，促進垃圾分類的自動化，並支持永續環境發展。準確的垃圾分類將有助於資源回收率的提升，減少環境污染，進而實現更為永續的生活方式。透過這樣的技術創新，將對全球環境保護做出積極貢獻，推動可持續發展目標的實現。

二、探究題目與動機

近年來，全球氣候變遷日益嚴重，而人類活動正是主要原因之一。臭氧層破壞、森林濫伐與垃圾過度生產，皆對環境造成巨大衝擊。為實現永續發展，除了政府的環保政策，企業與民眾的參與亦不可或缺。其中，最直接且有效的行動便是**改善垃圾分類**，確保資源能被最大化利用。傳統垃圾處理以掩埋為主，但隨著人口增長與土地資源有限，這種方式難以長期維持。焚化雖能減少垃圾體積，卻可能釋放有害氣體，且錯誤分類導致可回收資源被焚毀，造成浪費。因此，**精確、高效的垃圾分類**至關重要。

然而，現行分類方式效率低落，執行困難。我們意識到，若能運用人工智慧與機器學習技術，**自動識別**與分類垃圾，將大幅提升**資源回收率**，減少環境負擔。因此，我們決定結合研究與實作，以科技提升垃圾分類效率，為永續環境盡一份心力。

本計畫將透過設計一款垃圾辨識模型，實際搭配鏡頭先運用在協助班級上的垃圾分類，未來在進一步拓展應用到公共場合，例如：圖書館、捷運站、夜市等設有垃圾桶的場所，以落實垃圾的分類，實現**永續環境目標**之一的資源再利用。

三、探究目的與假設

(一) 實作的研究架構：



圖 1：研究流程架構圖

(二) 文獻探討和初步假設：

1. 各種程式語言的比較

當我們在撰寫程式時，程式語言本身是一個很大的影響因素，這其中考慮到許多面向：實用性 (Usability)、效能 (Performance)、等等，因此不同種類的程式執行的結果也會有所差異。因此我們比較各種不同的程式語言的差異、優缺點等；如下表 3 所示。

表 1：程式語言比較表

	Python	C++	Javascript	Java
上手難度	低	高	中	中
語法特徵	簡潔易讀	高效	動態交互	面相對象
大眾趨勢	高	高	低	低
使用場景	基礎工程師、資料擷取、AI 人工智慧工程師	遊戲開發、系統程式	Web 前端、Web 後端、遊戲開發	大型應用程式、Android App 開發

2. DenseNet 深度神經網路

DenseNet (密集連接卷積神經網路) 是一種改進型的深度學習網路架構，由 Gao Huang 等人於 2017 年提出。該架構主要應用於圖像分類、物件偵測和影像分割等電腦視覺任務。相較於傳統卷積神經網路 (CNN)，DenseNet 透過「密集連接」的方式—即每一層的輸出都會直接連接到後續所有層，而不僅僅是傳遞到下一層，這與傳統的 ResNet (殘差網路) 有所不同—提高特徵重用效率，減少參數量，並緩解梯度消失問題，使得模型在相同計算資源下達到更優越的表現。

(1) 特徵重用

DenseNet 通過特徵重用開發了網路的潛力，產生易於訓練且參數效率高的壓縮模型。將不同層學習的特徵圖連接起來，再後續層輸入的變化提高效率。

(2) 梯度流動

梯度消失問題 (Vanishing gradient problem) 是一種存在於機器學習中的難題。在每次

訓練中，神經網路的值與函數的導數應成比例，然而在某些情況下，梯度值會幾乎消失，使之無法得到有效更新，甚至可能完全無法繼續訓練。簡單來說，就是隨著資訊的傳遞，其中的資料可能會逐漸或突然消失；舉個例子：所有人站成一排由第一個人向最後一個人喊話，隨著隊伍越拉越長聲音會漸漸變小、直到聽不見。而 DenseNet 的密集連接可以頻繁的使梯度流動，讓梯度消失無法發生。

(3)密集連接

「其中每個層都以以前向方式直接與其他所有層連接（在每個 Dense block 內部）。對於每一層，它前面所有層的特徵圖都被視為單獨的輸入，而其自身的特徵圖則傳遞給所有後續層作為輸入。」(Gao Huang & Zhuang Liu & Laurens van der Maaten & Kilian Q. Weinberger, 2017)。簡單來說就是假設有 A,B,C,D 三層，而 A 層會連接到 B 和 C 層、B 層會連接到 C 和 D 層、則 C 層會連接到 D 層。密集連接是 DenseNet 的特色之一，他可以有效地減輕梯度消失問題、減少參數量、造成特徵重用；但是缺點它的運行速度會較一般的模型稍慢。

四、探究方法與驗證步驟

(一) 程式編寫

部分程式編寫展示，如圖 2、3 所示。

```
[ ] %cd "/content/drive/MyDrive/專題/garbage-dataset/test"
/content/drive/MyDrive/專題/garbage-dataset/test

def process_image(image):
    # Get the height and width of the image
    h, w, _ = image.shape
    aspect_ratio = w / h

    # If the aspect ratio is greater than 1.25, pad the top and bottom to adjust the aspect ratio to 1
    if aspect_ratio > 1.25:
        # Calculate the new height to adjust the aspect ratio to 1
        new_h = w
        padding = (new_h - h) // 2

        # Use the same padding for the top and bottom, and copy the edges to fill
        image = cv2.copyMakeBorder(image, padding, padding, 0, 0, cv2.BORDER_REPLICATE)

    # If the aspect ratio is less than 0.8, pad the left and right to adjust the aspect ratio to 1
    elif aspect_ratio < 0.8:
        # Calculate the new width to adjust the aspect ratio to 1
        new_w = h
        padding = (new_w - w) // 2

        # Use the same padding for the left and right, and copy the edges to fill
        image = cv2.copyMakeBorder(image, 0, 0, padding, padding, cv2.BORDER_REPLICATE)
```

圖 2：部分程式內容展示

```
img = cv2.resize(image, (224, 224))
return img

import numpy as np
import tensorflow as tf
import cv2
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt
import pandas as pd

img = cv2.imread('test_2.jpg') # 讀取圖片
img = process_image(img)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # 將 BGR 轉換為 RGB
plt.imshow(img_rgb) # 顯示圖片

img_array = np.array(img) # 轉換為 NumPy 陣列
img_array = img_array / 255.0 # 正規化數據
img_array = np.expand_dims(img_array, axis=0) # 增加 batch 維度
```

圖 3：部分程式內容展示

(二) 實際架設

完成了模型建構後，並測試基本運行沒有問題後，我們將其接上開發板和鏡頭，並將其架設在垃圾桶之上，如下圖 4。



圖 4：實際架設圖

鏡頭的位置如圖 5 所示，鏡頭會及時回傳視訊畫面，並自動擷取成圖片導入模型運行，以獲得即時的垃圾類型回饋。

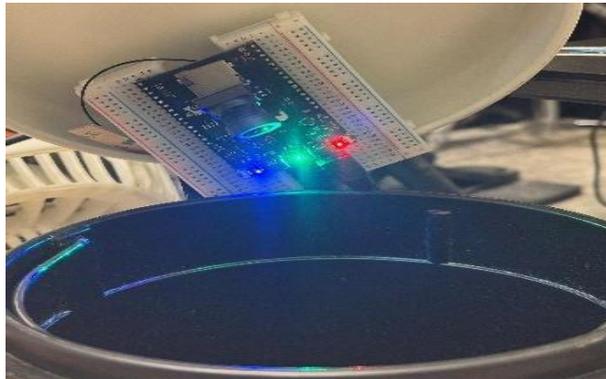


圖 5：實際架設圖

(三) 實際測試和分析

以下是部分垃圾種類實際偵測後的運行成果展示，如圖 6、7、8。

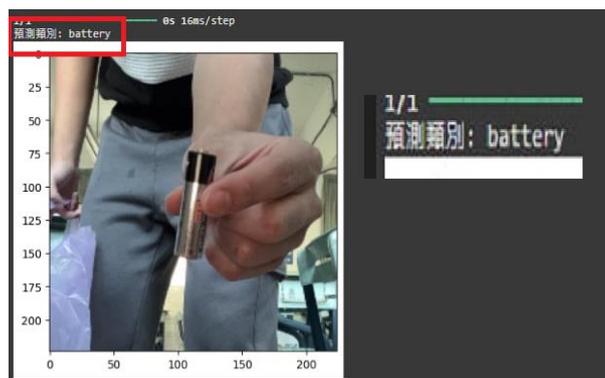


圖 6：電池類實際偵測展示

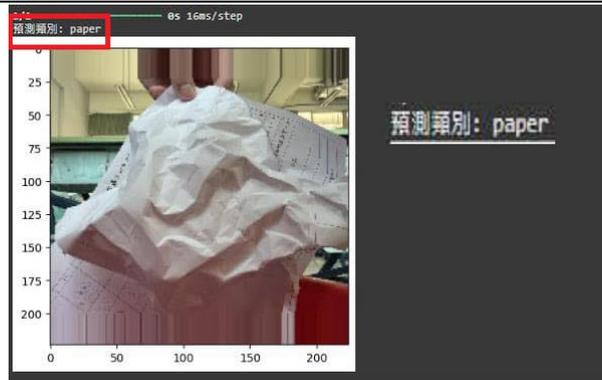


圖 7：紙類實際偵測展示



圖 8：塑膠垃圾類實際偵測展示

五、結論與生活應用

(一) 測試統計表

為了進行大數據的分析，我們除了把原本手上有的實體垃圾全部分析完後，在網上找了其他的垃圾圖片給鏡頭進行比較。最後我們統計了垃圾辨識的結果與數量，並做成圖表如下：

表 8：垃圾辨識結果

	塑膠垃圾	電子類垃圾(含電池)	紙類	完整玻璃類	鐵鋁類
成功數量	224	250	254	302	179
失敗數量	46	16	66	3	106
總計	270	266	320	305	285
成功率	82.9%	94%	79.2%	99.1%	62.6%

表 8 資料來源：研究者自行彙整分析(四捨五入到小數點後第 3 位)

(二) 結果分析

根據上表 8 我們可以得出以下結論並推斷理由：

1. 成功和失敗率

在所有的垃圾中成功率最高的垃圾種類是完整玻璃類，其次是電子垃圾類；最差的分別是鐵鋁和紙類。我們推斷可能原因如下：

(1) 完整垃圾：因為我們選擇的玻璃和電子垃圾類的圖片都不是有殘缺的(例如缺一腳的碎玻

璃杯)·所以比起皺的紙類和鐵鋁類會有更佳的辨識結果。

(2)失敗率：鐵鋁類成功率較低的原因·除了提供給電腦的圖片資料不完整外圖片反光的部分也是造成無法判別和辨識錯誤的原因。

2.如何改善

根據我們對上述的辨識成功和失敗的原因進行妥善分析後·我們認為有幾項步驟可以進行改善：

(1)把輸入給電腦的資料圖片進行更清楚的處理像是去反光等

(2)增加不完整垃圾圖片的量

(三)生活應用

目前的模型能達成基礎的分類功能·我們目前將其運用在**家庭垃圾桶**和**班級垃圾桶**的檢測上·一旦有錯誤分類的情形發生·模型就會提出警示·以此改善分類的情況。未來我們希望透過增加資料庫·提升模型行能以達到更好的辨識**精確度**和**效率**·進一步將此自動垃圾辨識系統**拓展應用到公共場合**·例如：圖書館、捷運站、夜市。

參考資料

Gao Huang & Zhuang Liu & Laurens van der Maaten & Kilian Q.Weinberger (2017) . Densely Connected Convolutional Networks.

葉欣睿 (譯) (2019) 。 Deep learning 深度學習必讀：Keras 大神帶你用 Python 實作。旗標出版社。

Saleh A. Albelwi (2022) . Deep Architecture based on DenseNet-121 Model for Weather Image Recognition. *International Journal of Advanced Computer Science and Applications*.

洪錦魁 (2022) 。 Python 最強入門邁向頂尖高手之路：王者歸來(第二版)全彩版。深智數位出版社。

陳昭明 (2024) 。深度學習 最佳入門邁向 AI 專題實戰 (二版) 。深智數位出版社。

《A logical calculus of the ideas immanent in nervous activity》WARREN S. MCCULLOCH
AND WALTER PITTS

《 The Organization of Behavior 》 B. F. SKINNER ASSISTANT PROFESSOR OF
PSYCHOLOGY