

2025 年【科學探究競賽-這樣教我就懂】

普高組 成果報告表單

題目名稱：探討大語言模型於協助照顧植物之可行性與效果

一、摘要

本研究因應環境的破壞與汙染，為響應環保以及愛護地球等行動，將協助人們了解並種植栽盆作為動機。本研究透過未經微調的 Gemini 1.5 Flash 模型協助沒有經驗者種植植物，並以網頁服務的開發與實際測試，驗證大語言模型對於植物品種、疾病辨識與敘述種植環境等能力。雖辨識疾病方面在本研究中無法確認是否已有可商業化的準確率，但辨識植物品種已透過足夠廣的樣本，驗證其可投入實際使用的準確率。為更深入驗證可行性，本研究呼籲投入更多資源研究並發展大語言模型及相關人工智能在植物照顧協助的潛力。

二、探究題目與動機

隨著人工智慧技術的蓬勃發展，大型語言模型（Large Language Model）已成為推動各產業創新的重要科技。因此，本研究希望藉由新興科技，為環保盡一份心力。Google 所開發的 Gemini 模型，以其卓越的語言理解能力和開放的 API 介面，為研究者和開發者提供了強大的工具。儘管現今大型語言模型在圖像理解和分析方面已取得顯著進展，但在園藝領域的應用，特別是植物品種辨識和照護方面，仍有待深入探索。

本研究旨在透過 Python 程式語言，結合 Gemini API，對其在園藝相關應用進行系統性的評估與實作。透過對 Gemini 在植物品種辨識、病蟲害診斷以及提供照護建議等方面的成效評估，本研究期望能深入探索大語言模型在園藝領域的應用潛力。因此，本研究將致力於開發基於 Gemini API 的園藝輔助工具，以降低園藝門檻，協助無經驗者進行植物種植與照護。此舉不僅能推廣永續園藝，更能鼓勵更多人參與植物種植，為環境保護貢獻一份力量。本研究期望透過對 Gemini 在園藝領域的應用探索，不僅能評估其技術潛力，更能開發出實用的工具，促進永續園藝的普及，為環境保護盡一份心力。

三、探究目的與假設

研究目的：

- （一）探索大語言模型於植物品種辨識的能力。
- （二）探索大語言模型於植物疾病辨識的能力。
- （三）探索大語言模型於植物照顧協助的能力。

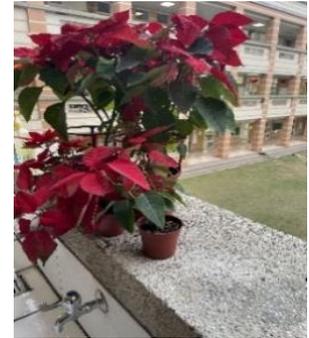
研究假設：

- （一）Gemini API 可接收並交由 Gemini 1.5 Flash 理解圖片內容並對其進行分析。
- （二）Gemini 1.5 Flash 與市面上流行的大語言模型有類似的分析能力。
- （三）使用者日常上傳的照片特質包含：整株且單株植物、畫面清晰、曝光充足不過量。
- （四）擁有相同常見名稱的植物將擁有相似的生長習性及外表。

四、探究方法與驗證步驟

(一) 研究方法

本研究在 Python 程式中整合 Gemini API，利用 Flask 架構呈現網站，並測試與驗證大語言模型在園藝相關領域的應用。為此，筆者首先透過 Gemini API 與程式碼調用 Gemini 1.5 Flash 進行簡單的問答：上傳圖片的模型，並設置提示詞，使其進行植物判斷，並回復植物的學名以及常見名。獲得植物的學名後，利用三個不同提示詞的模型，包括：植物疾病、視覺判斷和植物生長條件模型，進行延伸功能。為了模擬並測試程式的實際使用，以 50 張各式尺寸、解析度、檔案格式、光影、植物的圖片將透過 Gemini API 上傳至模型。



【圖一】範例

【圖一】為範例圖片。

(二) 網頁程式開發

本研究先透過基礎功能的開發，核實概念的可行性。並藉此深入研究 Gemini API 未經過微調 (Fine Tuning) 是否能直接透過圖片判斷植物並診斷疾病。

1. 首先，透過【圖二】中展示的 Python 程式碼，調用 Gemini 1.5 Flash 模型，進行簡單儲存並上傳。

```
10 def save_uploaded_file(uploaded_file):
11     if uploaded_file:
12         file_extension = os.path.splitext(uploaded_file.filename)[1]
13         save_dir = "HomeGardener/App/static/user_upload_files"
14
15         try:
16             os.makedirs(save_dir, exist_ok=True)
17             print(f"Directory {save_dir} created successfully or already exists.")
18         except Exception as e:
19             print(f"Failed to create directory {save_dir}: {e}")
20             return None
21
22         save_path = os.path.join(save_dir, uploaded_file.filename)
23
24         try:
25             uploaded_file.save(save_path)
26             print(f"File saved successfully at {save_path}")
27             return save_path
28         except Exception as e:
29             print(f"Failed to save file {uploaded_file.filename}: {e}")
30             return None
31     return None
```

【圖二】Python 程式碼。

2. Gemini 1.5 Flash 模型透過 Gemini API 獲取圖片後，將回傳值處理、分割並分別存至【圖三】中的不同變數：植物學名 (species)、常見名 (common_name) 與疾病診斷 (disease)。

```

plt_ide_dis_dec = plt_recog_dis_dect(image_path, api)
print(plt_ide_dis_dec)
if not plt_ide_dis_dec or len(plt_ide_dis_dec) < 2:
    raise ValueError("Invalid response from plant_recognition API")

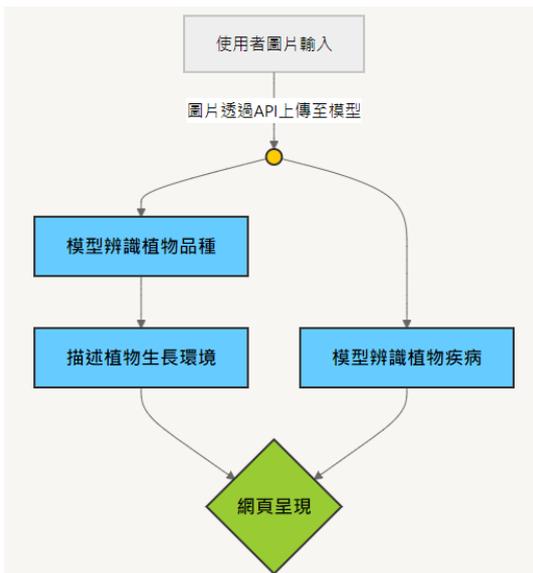
plant = plt_ide_dis_dec[0]
disease = plt_ide_dis_dec[1]
species_name = plant[0]
common_name = plant[1]

growing_condition = gen_rec(species_name, api)
if "0" not in disease:
    disease_management = dis_man(species_name, disease, api)
else:
    disease_management = "There is no disease present"

```

【圖三】將 Gemini 模型的回傳結果進行儲存

3. 透過 Flask 架設網站，並在 HTML 網頁中，呈現內容。【圖四】展示流程；【圖五】展示網頁畫面。



【圖四】網頁架構流程圖

Plant Information

Plant Image



Plant Info

Species Name: Kalanchoe blossfeldiana

Common Name: Kalanchoe

Growing Condition

Bright, indirect light; Well-draining potting mix (cactus & succulent blend); Water thoroughly only when soil is completely dry; Balanced liquid fertilizer diluted to half strength, monthly during growing season, 65-75°F (18-24°C); Average household humidity is fine.

Disease Management

There is no disease present

【圖五】網頁內容呈現

(三) 資料清理

在已有研究的架構後，還需要透過日常植物照片樣本進行測量精準度，以確保準確性與實用性。首先，為了因應使用者上傳圖片的多樣性，在以上假設下，使用 50 張以單株植物為主角的測試資料，進行了以下步驟測試程式的效能：

1. 將圖片上傳至模型，並在 log.csv 記錄植物辨識與疾病辨識結果。log.csv (展示於【表一】) 包含以下數值：檔案名稱 (filename)、植物名 (plant name)、植物辨識 (1 表示正確；0 表示錯誤)、疾病辨識 (disease recognition)。

註：判斷植物因面對大眾使用，因此不須精確至「種」，若常見名相同，並擁有相似特徵的植物也會被紀錄為正確。

【表一】log.csv 的前三筆資料

filename	plant name	plant recognition	disease recognition
1739685212_areca.jpg	Areca palm	1	0
1739685327_Bald_cypress2.jpg	Cypress	1	0
1739685375_cannabis.jpg	Cannabis	1	0

2. 為了使資料分析更容易，每筆資料中，透過人工判斷的方式，添增疾病 (diseased) 一欄。再與疾病判斷 (disease recognition) 結合，整理出判斷是否正確，儲存於 correct 一欄中。【表二】展示 log.csv 的前三筆資料。

【表二】log.csv 的前三筆資料

filename	plant name	plant recognition	disease	correct	disease recognition
1739685212_areca.jpg	Areca palm	1	0	1	0
1739685327_Bald_cypress2.jpg	Cypress	1	0	1	0
1739685375_cannabis.jpg	Cannabis	1	0	1	0

(四) 計算植物辨識準確率

原資料中，將 1 設為正確判斷，0 設為錯誤判斷，因此在分析時，只需取得植物判斷的平均值，即可算出整體資料的正確率。筆者透過 Numpy 的 mean() 函數計算平均值，亦植物判斷的準確率為 **86%**。

(五) 計算植物健康判斷準確率

為了測量判斷疾病的準確率，將辨識結果分類至【表三】所展示混淆矩陣：真陽性 (TP)、偽陽性 (FP)、真陰性 (TN)、偽陰性 (FN)：

【表三】混淆矩陣

名稱	True Positive	True Negative	False Positive	False Negative	Diseased Plant	Healthy Plant
數值	4	36	8	2	6	44

取真陽性與真陰性之合處已全部的合即可計算出準確率： $(4 + 36) \div (6 + 44) = 80\%$

若只取有疾病的植物計算準確率： $4 \div 6 = 66.7\%$

(六) 植物生長條件

透過 Python 程式碼使用 Geminin API 調用 Gemini 1.5 Flash 向使用者提供植物生長的環境，以協助使用者對植物進行照顧。

```
for api in api_list:
    try:
        plt_ide_dis_dec = plt_recog_dis_dect(image_path, api)
        if not plt_ide_dis_dec or len(plt_ide_dis_dec) < 2:
            raise ValueError("Invalid response from plant_recognition API")

        plant = plt_ide_dis_dec[0]
        disease = plt_ide_dis_dec[1]
        temp_species = plant[0]
        temp_common = plant[1]
        temp_growing = gen_rec(temp_species, api)
        if "0" not in disease:
            temp_disease = dis_man(temp_species, disease, api)
        else:
            temp_disease = "There is no disease present"
```

【圖六】植物照顧及生長環境生成

其中【圖六】的 gen_rec() 函數接收植物學名 (temp_species) 和 API 回傳植物的生長環境，最終於網頁加載時呈現於畫面。

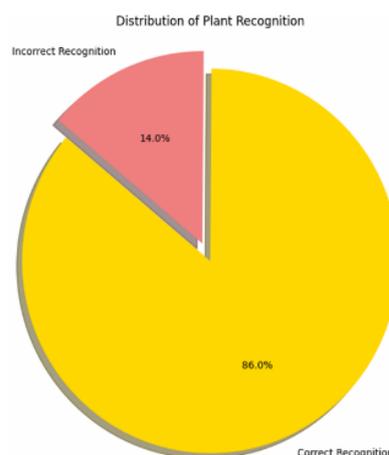
(七) 準確率分析

1. 植物品種辨識結果分析

大語言模型於判斷植物品種的準確率為 86% (呈現於【圖七】)，以符合預期的日常用途的準確率。然而，本研究的樣本圖片中，僅有 12% 的有疾病植物，因此，無法比較植物有無疾病對於大語言模型再判斷植物品種的影響。

註：**Incorrect Recognition** 為錯誤的品種判斷；

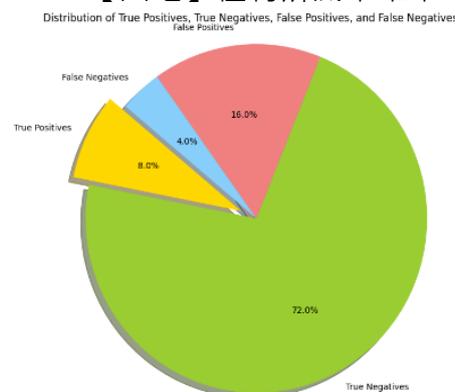
Correct Recognition 為正確的品種判斷。



【圖七】植物辨識準確率

2. 植物健康判斷結果分析

將每個數值除以總數 (50)，即可得到百分比。並計算真陽性與真陰性的合，亦是準確率 80%。然而若計算有疾病的植物的總數，即可發現僅占比所有圖片樣本的 12%，並且偽陰性佔有疾病的植物三分之一，因此若只計算有疾病的植物，準確率僅約 67%。由此總結疾病辨識的效能有潛力，但不能完全依賴並作為準確的判斷依據。【圖八】展示疾病辨識結果的混淆矩陣。



【圖八】混淆矩陣

五、結論與生活應用

- (一) 大語言模型在未經微調的日常植物圖片中，判斷植物品種準確率可達 86%。
- (二) 大語言模型在未經微調的日常植物圖片中，判斷植物是否健康準確率可達 80%；然而，此結論須更多研究佐證，因此呼籲其他研究者的參與。
- (三) 大語言模型有能力生成植物的生長條件，協助無經驗者種植栽盆；然而，其正確與否仍需更多研究參與核實。

參考資料

1. Amazon Web Services, Inc. (2021). What is Deep Learning? - Deep Learning AI Explained - AWS. Available at: <https://aws.amazon.com/what-is/deep-learning/>
2. Amazon Web Services, Inc. (2023). 什麼是 LLM ? – 大型語言模型說明 – AWS. Available at: <https://aws.amazon.com/tw/what-is/large-language-model/>
3. European Food Safety Authority. (2025). The role of plants. Available at: <https://www.efsa.europa.eu/en/plh4l/role-plants>
4. Giron, S. (2024). Gemini 1.5 Pro and Flash reasoning capacities versus ChatGPT 4o and OpenAI o1. Medium. Available at: <https://medium.com/@stephane.giron/gemini-1-5-pro-and-flash-reasoning-capacities-versus-chatgpt-4o-and-openai-o1-1035a9ffa580>
5. Google AI for Developers. (2025). Gemini models. Available at: <https://ai.google.dev/gemini-api/docs/models/gemini#gemini-1.5-flash>.
6. Palletsprojects.com. (2025). Welcome to Flask — Flask Documentation (3.1.x). Available at: <https://flask.palletsprojects.com/en/stable/>
7. Twsgi.org.tw. (2015). 台灣創價學會 TSA、Taiwan Soka Association. Available at: https://www.twsgi.org.tw/index-news-detail.php?n_id=5106
8. Yeh, S. (2020). Python Web — 快速建置網站的 Flask 套件 - Sean Yeh - Medium. Available at: <https://seanyeh.medium.com/python-web-%E5%BF%AB%E9%80%9F%E5%BB%BA%E7%BD%AE%E7%B6%B2%E7%AB%99%E7%9A%84flask%E5%A5%97%E4%BB%B6-59318830bd63>