指導教授 林仁祥

辨識國旗

第三組

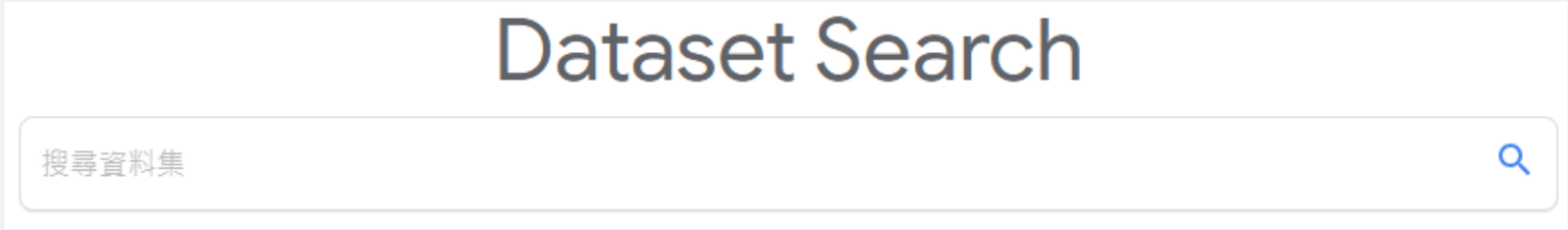
電機三 翁得恩 411500506

電機三 楊舒凱 411506529

目錄

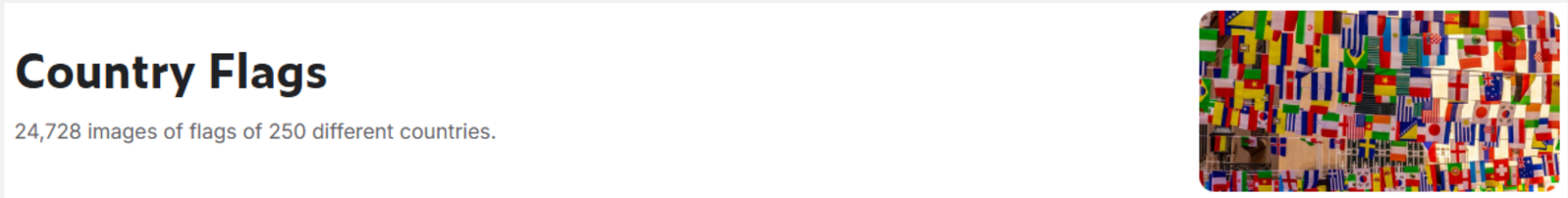
資料來源

- Google Dataset Search



https://datasetsearch.research.google.com/search?src=0&query=flag&docid=L2cvMTFzdGtmd2g1cw%3D%3D

- Kaggle



https://www.kaggle.com/datasets/jacobpatton/country-flags

資料處理

掛載雲端硬碟

```
from google.colab import drive
import zipfile
import os

drive.mount('/content/drive')  # 掛載 Google 雲端硬碟

zip_path = '/content/drive/My Drive/flags.zip'  # 指定Zip檔案路徑
extract_path = '/content/dataset'  # 解壓縮目標資料夾

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
        zip_ref.extractall(extract_path)  # 解壓縮 .zip 檔案
```

數據處理與準備

```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.applications import ResNet50
import numpy as np
import os
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

train_path = "/content/drive/MyDrive/Colab Notebooks/dataset/flags/train" # 訓練集路徑
test_path = "/content/drive/MyDrive/Colab Notebooks/dataset/flags/test" # 測試集路徑
image_size = (224, 224)
batch_size = 32
```

資料處理

- 載入數據集

```
train_dataset = tf.keras.utils.image_dataset_from_directory(
  train_path,
  image_size=image_size,
  batch_size=batch_size
)

test_dataset = tf.keras.utils.image_dataset_from_directory(
  test_path,
  image_size=image_size,
  batch_size=batch_size
)
class_names = train_dataset.class_names
```

資料處理

- 數據增強

V1

```
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.2),
])
```

V2
V3

```
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal_and_vertical"),
    layers.RandomRotation(0.3),
    layers.RandomZoom(0.2),
    layers.RandomContrast(0.2),
])
```

V4

```
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal_and_vertical"),
    layers.RandomRotation(0.3),
    layers.RandomZoom(0.3),
    layers.RandomContrast(0.3),
    layers.RandomBrightness(0.2),
])
```

資料處理

預處理

```
def preprocess(image, label):
    image = data_augmentation(image)
    image = tf.cast(image, tf.float32) / 255.0 # 正規化到 [0, 1]
    return image, label
```

V1

```
train_dataset = train_dataset.map(preprocess)
test_dataset = test_dataset.map(preprocess)
```

V2

V3

V4

```
train_dataset = train_dataset.map(preprocess)
test_dataset = test_dataset.map(lambda x, y: (tf.cast(x, tf.float32) / 255.0, y))
```

模型製作流程

模型建構

V1

```
預訓練模型 ResNet50
數據集訓練權重 ImageNet
base_model.trainable = False
GlobalAveragePooling2D()
layers.Dense(256, activation="relu")
layers.Dropout(0.5)
layers.Dense(num_classes, activation="softmax")
```

V2 V3 V4

```
base_model.trainable = True
解凍 ResNet50 的最後 50 層
layers.Dense(256, activation="relu", kernel_regularizer=tf.keras.regularizers.l2(0.01))
```

模型製作流程

- 編譯模型

V1

```
model.compile(
    optimizer="adam",
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)
```

V2 **V3** **V4**

```
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)
```

模型製作流程

- 模型訓練_每代保留原有功能 增加新功能

V1

epochs 10次

V2

模型檢查點保存 ModelCheckpoint
早停策略 EarlyStopping
動態學習率調整 ReduceLROnPlateau
epochs 20次

V3

從現有模型繼續訓練 load_model
僅保存驗證損失最低的模型 save_best_only=True monitor="val_loss"
epochs 30次

V4

增加混淆矩陣、分類報告
epochs 10次

模型製作流程

- 模型訓練結果

V1

accuracy: 0.0578 - loss: 4.6308 - val_accuracy: 0.1500 - val_loss: 4.2935

V2

accuracy: 0.5257 - loss: 4.2205 - val_accuracy: 0.5984 - val_loss: 3.7408

V3

accuracy: 0.7162 - loss: 2.0536 - val_accuracy: 0.7324 - val_loss: 1.9277

V4

accuracy: 0.7112 - loss: 1.8797 - val_accuracy: 0.7256 - val_loss: 1.7526

功能內容

- 定義預測函數

```
def predict_flag(image):
 # 預處理圖片
     image = image.resize((224,  224)) # 確保大小與模型輸入匹配
     image = np.array(image) / 255.0 # 正規化
     image = np.expand_dims(image, axis=0) # 增加批次維度

 # 模型預測
     predictions = model.predict(image)
     predicted_class = class_names[np.argmax(predictions)]
     confidence = np.max(predictions)

     return f"國旗: {predicted_class}, 置信度: {confidence:.2f}"
```

功能內容

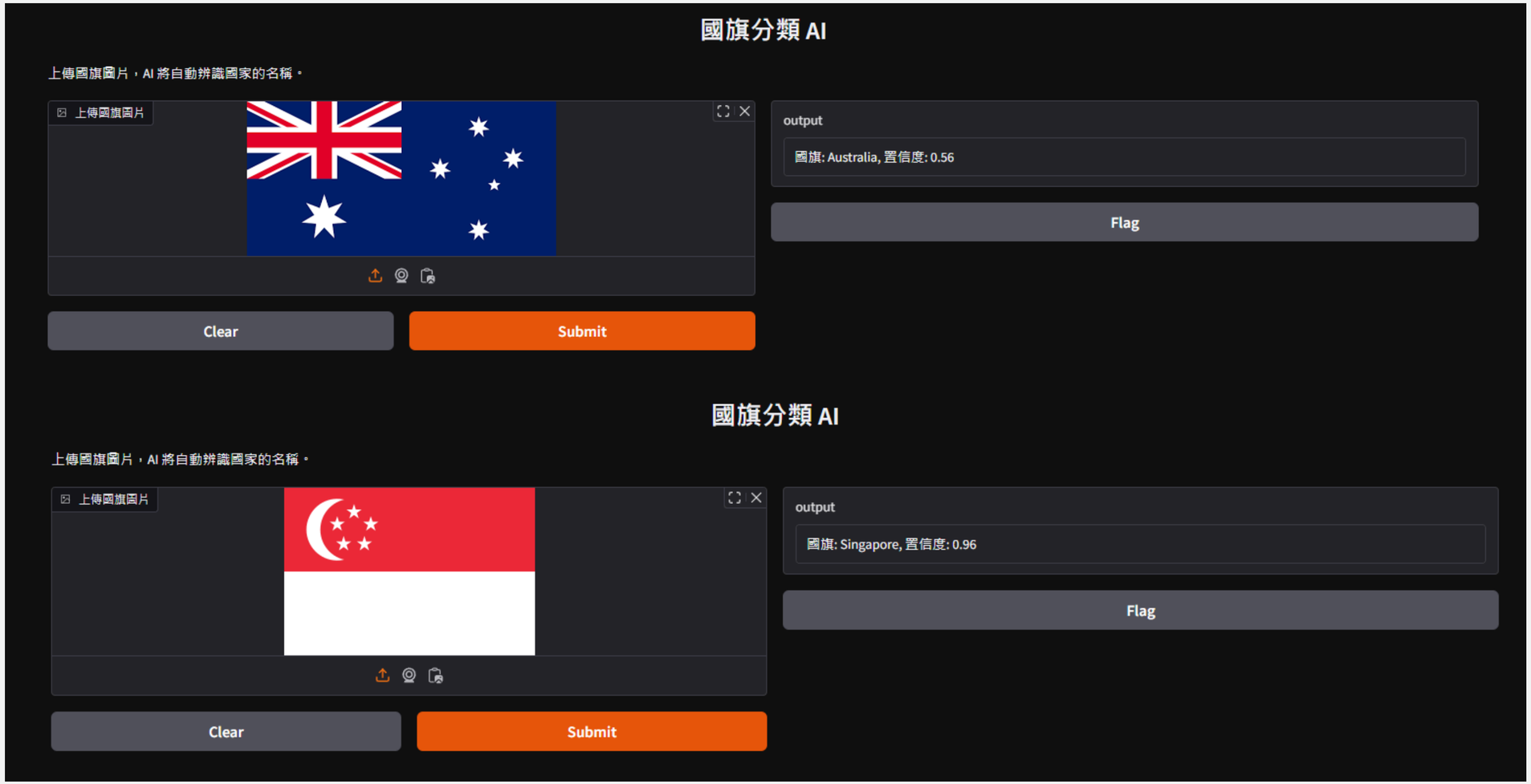
- 建立 Gradio 接口

```
interface = gr.Interface(
        fn=predict_flag,
        inputs=gr.Image(type="pil", label="上傳國旗圖片"),
        outputs="text",
        title="國旗分類 AI",
        description="上傳國旗圖片，AI 將自動辨識國家的名稱。"
)

interface.launch()
```

功能內容

- 功能內容


國旗分類 AI
上傳國旗圖片，AI 將自動辨識國家的名稱。
上傳國旗圖片
output
國旗: Australia, 置信度: 0.56
Flag
Clear
Submit
國旗分類 AI
上傳國旗圖片，AI 將自動辨識國家的名稱。
上傳國旗圖片
output
國旗: Singapore, 置信度: 0.96
Flag
Clear
Submit

功能內容

- 影片

組員分工

翁得恩

- 簡報製作
- 題目發想

楊舒凱

- 資料收集
- 程式撰寫

Colab程式碼

- 國旗辨識V1
- 國旗辨識V2
- 國旗辨識V3
- 國旗辨識V4
- 模型OutPutGradio

Thank you for listening!