
明星撞臉
影像辨識
★使用CNN★

宋芸樺、夏于喬是演藝圈有名的「撞臉明星」
藉由訓練CNN模型嘗試能否成功分辨兩人

網路爬蟲收集圖片

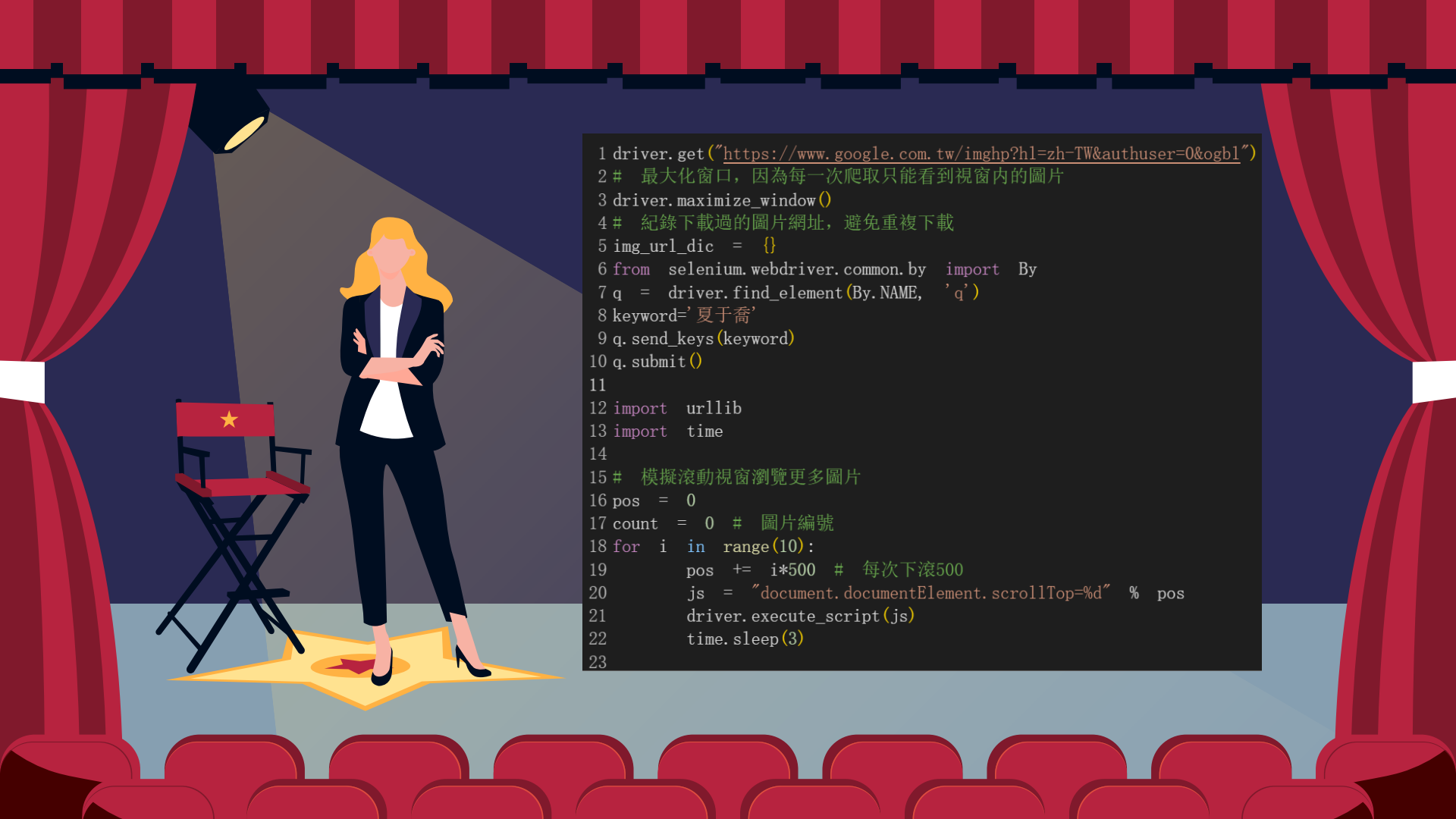
建立模型訓練/預測

圖片資料擴增

評估模型


01
爬取圖片
Selenium
Instaloader


Google
圖片



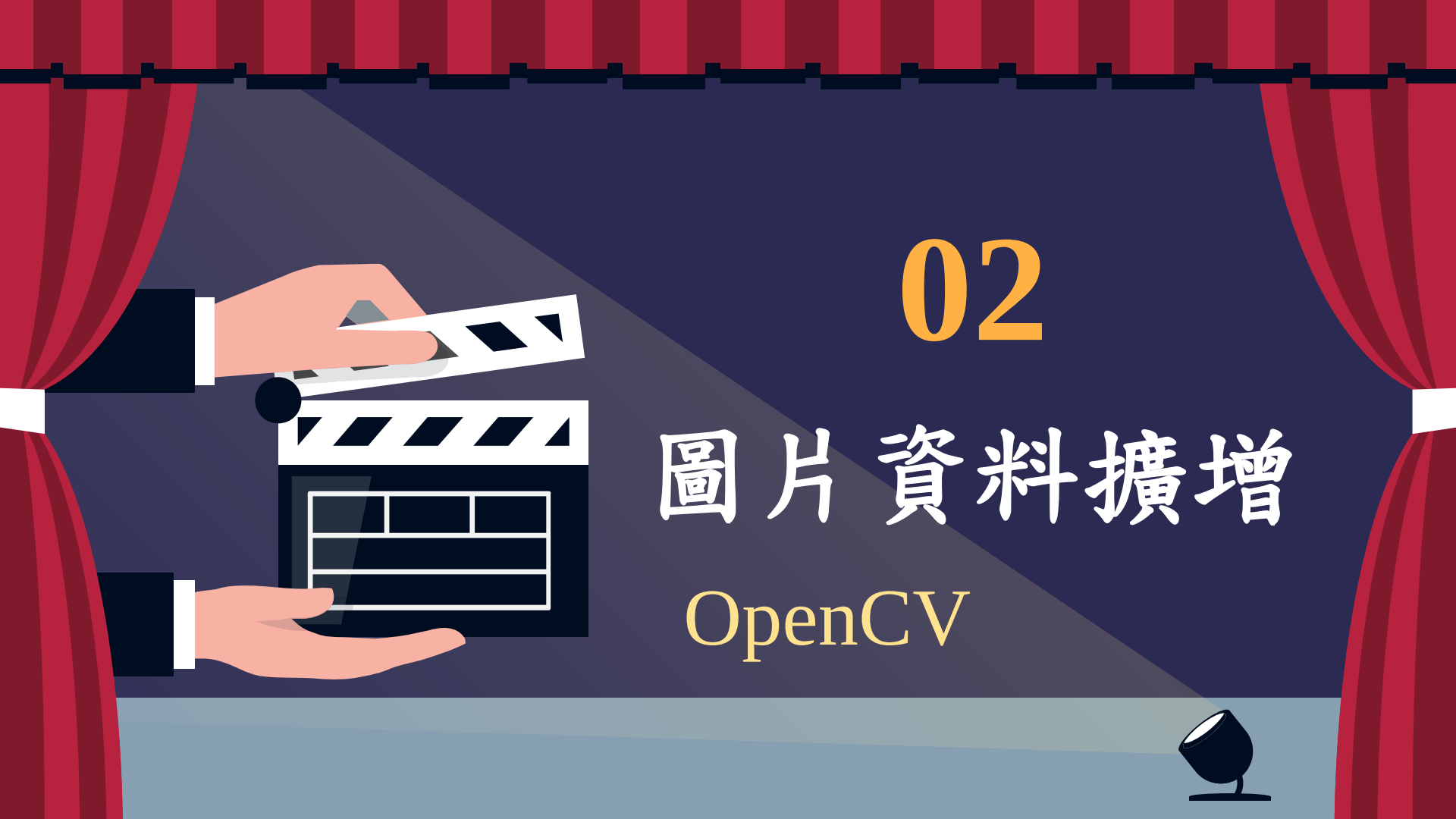
```python
driver.get("https://www.google.com.tw/imghp?hl=zh-TW&authuser=0&ogbl")
#  最大化窗口，因為每一次爬取只能看到視窗內的圖片
driver.maximize_window()
#  紀錄下載過的圖片網址，避免重複下載
img_url_dic  =  {}
from  selenium.webdriver.common.by  import  By
q  =  driver.find_element(By.NAME,  'q')
keyword='夏于喬'
q.send_keys(keyword)
q.submit()

import  urllib
import  time

#  模擬滾動視窗瀏覽更多圖片
pos  =  0
count  =  0  #  圖片編號
for  i  in  range(10):
        pos  +=  i*500  #  每次下滾500
        js  =  "document.documentElement.scrollTop=%d"  %  pos
        driver.execute_script(js)
        time.sleep(3)

```


instaloader / instaloader
Type / to search
Code
Issues 82
Pull requests 7
Actions
Security
Insights
instaloader
Public
Sponsor
Watch 152
master
8 Branches
127 Tags
Go to file
t
Code
aandergr Update sponsors in README
5509ea7 · last month
872 Commits
.github
Temporarily disable lint check on python 3.11
4 months ago
deploy
update CI dependencies, require Python>=3.8
last year
docs
update CI dependencies, require Python>=3.8
last year
instaloader
Release of version 4.10.3
last month
test
Improve typing syntax and type NamedTuples (#1610)
2 years ago
.gitignore
travis: use Pipfile for dependency pinning
5 years ago
.pylintrc
update CI dependencies, require Python>=3.8
last year
開源第三方套件Instaloader


02
圖片資料擴增
OpenCV

調整圖片大小

水平翻轉

資料擴增

將檔案中的圖片統一水平翻轉，擴展訓練集樣本

```python
import cv2
def data_additional(folder_path):
    image_files = os.listdir(folder_path)
    # 確定資料夾中只包含圖片檔案
    image_files = [file for file in image_files if file.endswith('.jpg')]
    n = 0
    # 處理每一張圖片，進行水平翻轉並保存到新的位置
    for file_name in image_files:
        # 讀取圖片
        n += 1
        image_path = os.path.join(folder_path, file_name)
        image = cv2.imread(image_path)

        # 水平翻轉圖片
        flipped_image = cv2.flip(image, 1)

        # 生成新的檔案名稱
        new_file_name = "flipped_" + file_name

        # 指定新檔案的保存路徑
        save_path = os.path.join(folder_path, new_file_name)

        # 將處理後的圖片保存到新的位置
        cv2.imwrite(save_path, flipped_image)
        print(f"水平翻轉並保存第{n}張圖片。")
```

03

模型建立

載入圖檔

圖像歸一化

建立模型

```
(train_images, train_labels), (test_images, test_labels) = load_data()
```

```
Loading train
100%|██████████| 808/808 [00:15<00:00, 51.58it/s]
100%|██████████| 802/802 [00:16<00:00, 47.53it/s]
Loading test
100%|██████████| 100/100 [00:01<00:00, 60.94it/s]
100%|██████████| 100/100 [00:01<00:00, 62.49it/s]
```

```
#隨機性
train_images, train_labels = shuffle(train_images, train_labels, random_state=25)
#標準化
train_images = train_images / 255.0
test_images = test_images / 255.0
```

```
input_shape = (150, 150, 3)

model = Sequential([
        Conv2D(50, (3, 3), input_shape=input_shape, padding='same',
                    activation='relu', strides=2),
        MaxPooling2D(pool_size=(2, 2), strides=2),
        Dropout(0.2),
        Conv2D(75, (3, 3), input_shape=input_shape, padding='same',
                    activation='relu', strides=2),
        MaxPooling2D(pool_size=(2, 2), strides=2),
        Dropout(0.2),
        Flatten(),
        Dropout(0.5),
        Dense(2, activation='softmax')  #輸出層，分類用softmax
])
```

編譯模型

EarlyStopping

載入模型訓練

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

```
from keras.callbacks import EarlyStopping

#  建立早停回調函數
early_stopping = EarlyStopping(monitor='val_accuracy', patience=5, restore_best_weights=True)
```

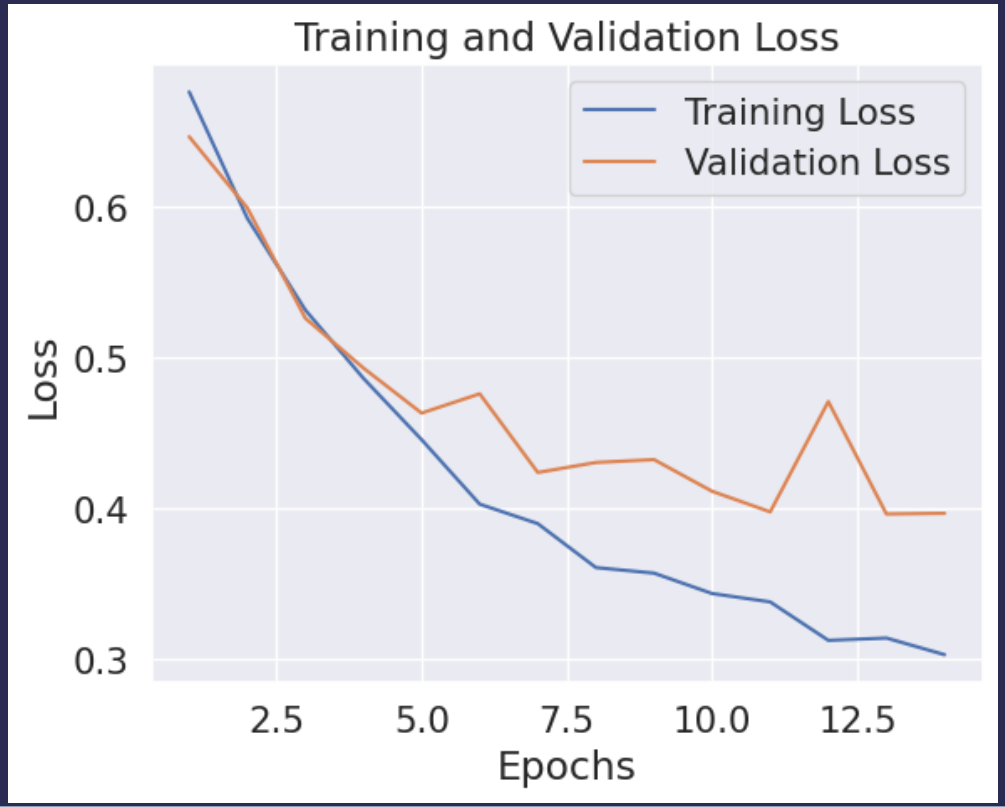
```
history = model.fit(train_images, train_labels,
                    batch_size=80, epochs=100,
                    validation_split=0.1, verbose=1,callbacks=[early_stopping])
```

Loss: 0.3027

Accuracy: 0.8737

Val loss: 0.3965

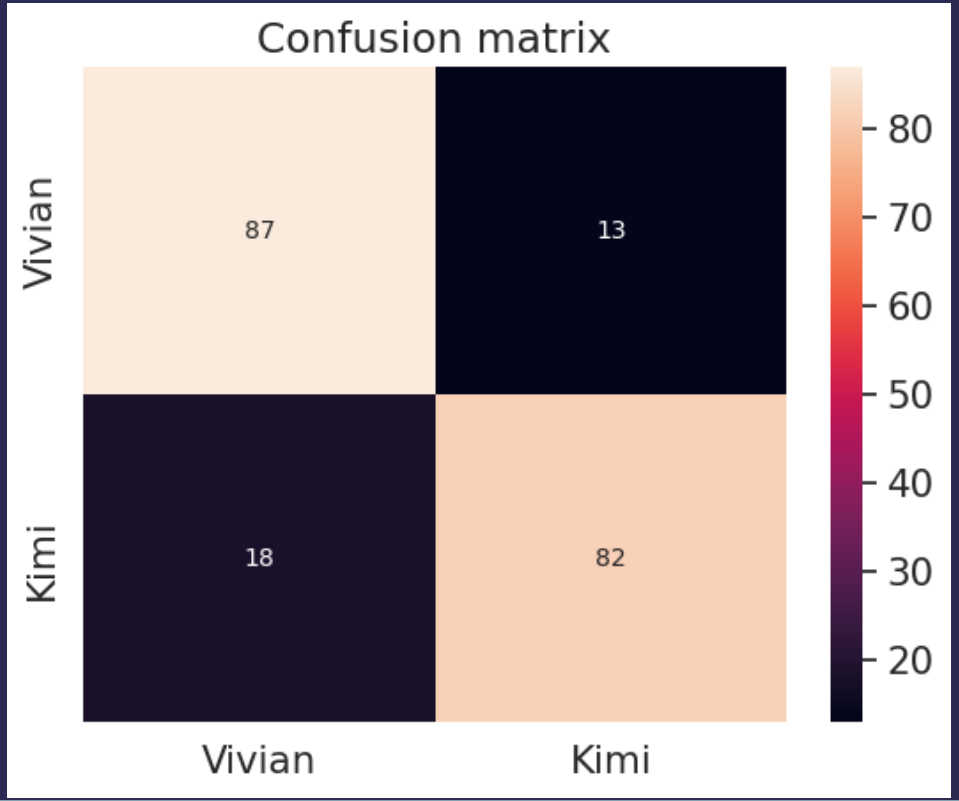
Val accuracy: 0.8509



模型預測測試集

Loss: 0.3729

Accuracy: 0.8450



	Train	Validation	Test
Loss	0.3027	0.3965	0.3729
Accuracy	87.37%	85.09%	84.50%

結論

訓練集和測試集損失函數跟精準度相差不大
沒有明顯的過擬合

增加資料集可能對提升模型準確度較有幫助


Thank You
For Watching!