

國立清華大學  
智慧化企業整合

Project 3

人眼辨識性別

第八組

110034555 李勁緯

指導教授:邱銘傳 博士

111/1/7

# 目錄

## 第一章、研究背景

1.1 研究背景說明 ..... 1

1.2 問題定義 (5W1H) ..... 1

## 第二章、文獻回顧 ..... 2

2.1 卷積神經網路 (Convolutional Neural Network, CNN) ... 2

## 第三章、研究架構及方法 ..... 3

3.1 資料來源 ..... 3

3.2 資料前處理 ..... 3

3.3 模型建立 ..... 5

3.5 訓練模型 ..... 6

## 第四章、結論與未來展望 ..... 8

## 第五章、參考文獻 ..... 9

# 人眼辨識性別

清華大學工業工程與工程管理學系 李勁緯

## 摘要

隨著貧富差距的擴大，竊盜、搶劫等案件層出不窮，犯罪者經常使用各種僅露出雙眼的偽裝技術，使警方難以辨識出其性別，大大增加了辦案的困難。因此，為了減輕警方辦案的困難，本研究建構一基於人工智慧的人眼辨識性別模型，藉由卷積神經網路辨識性別，能大幅減少警方偵查的範圍。

關鍵字：人眼辨識、卷積神經網路

# 第一章、研究背景

## 1.1 研究背景說明

金融風暴、疫情肆虐等影響經濟的發展，世界各地有很多人就此失業，貧富差距也越來越大。因此，竊盜、搶劫等案件層出不窮，犯罪者也經常使用各種僅露出雙眼的偽裝技術，使警方難以掌握其身分或甚至是性別，大大增加了辦案的困難。本研究將以 CNN model 建構人眼辨識性別模型，協助警方藉由人眼照片辨識性別，能大幅減少警方偵查的範圍，並快速的將犯罪者繩之以法。本研究使用卷積神經網路（Convolutional Neural Networks，CNN）深度學習的方法，並建立一人眼辨識性別模型。

## 1.2 問題定義（5W1H）

針對此研究主題，5W1H 分析說明如下：

Who：警方、相關偵查單位

What：犯罪者以各種僅露出雙眼的偽裝技術，使警方難以辨識出其性別

Why：協助警方辦案

When：蒐集到犯罪者的眼睛照片時

Where：世界各地

How：以 CNN 深度學習之方法建立人眼辨識性別模型

## 第二章、文獻回顧

### 2.1 卷積神經網路 (Convolutional Neural Network, CNN)

卷積神經網路 (Convolutional Neural Networks, CNN) 是一深度學習的演算法，可用於電腦視覺自動辨識、影像分類等應用上，是一以標籤資料進行模型訓練與驗證的監督式學習法。CNN 卷積神經網路模型，訓練完成後可以藉由擷取與影像形狀相關的特徵進行影像辨識 (Ketkar & Santana, 2017)。CNN 模型會從數據中自行學習特徵的識別，有別於過往的傳統影像識別方法，需要密集的進行特徵工程。CNN 的基本結構會包含多個交替堆疊的卷積層以及池化層，卷積層會使用多個濾鏡(kernels) 幫助模型擷取圖像特徵，這種構造使 CNN 架構在參數的使用上更有效率，也讓 CNN 架構的神經網路有高準確度的表現。而池化層可以降低特徵圖的維度讓網路的計算複雜度降低，可以更快的得到運算結果。本研究將使用基於卷積神經網路架構的模型實作圖像分類器，以達成人眼辨識性別。

## 第三章、研究架構及方法

### 3.1 資料來源

由 Kaggle 公開數據集中取得人眼的圖像資料集，本次為使用「Eye Classification」，分為男眼、女眼 2 種類別，男眼共 6323 張照片，女眼共 5202 張照片，總共 11525 張照片。

### 3.2 資料前處理

將資料集存放雲端，從 Colab 讀取雲端檔案。

```
from google.colab import drive
drive.mount('/content/drive')
```

插入 tensorflow，matplotlib 等常見第三方套件與模組。

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
```

設定參數，訓練跑幾 10 次、每批資料量的大小為 50、將每張圖片的大小統一設定為 256 X 256 的圖像，並載入資料集。

```
epochs = 10
batch_size = 50
image_size = (256, 256)
base_path = '/content/drive'
images_f = '/content/drive/MyDrive/femaleeyes'
images_m = '/content/drive/MyDrive/maleeyes'
seed = 82
```

將數據拆分為訓練集和驗證集，比例為 8：2，image\_dataset\_from\_directory 這個函數，它會讀取目錄中的檔案，存入 dataset，一次只會讀取一批 (batch) 資料。

```

train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    base_path,
    validation_split = 0.2,
    subset = "training",
    seed = seed,
    image_size = image_size,
    batch_size = batch_size,
)
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    base_path,
    validation_split = 0.2,
    subset = "validation",
    seed = seed,
    image_size = image_size,
    batch_size = batch_size,
)

```

```

Found 11525 files belonging to 2 classes.
Using 9220 files for training.
Found 11525 files belonging to 2 classes.
Using 2305 files for validation.

```

圖片大小設定為 10 X 10，每次從訓練集取一張照片，設定子圖為 3 行、3 列、圖片位置每次加 1，並利用 `imshow` 顯示訓練集中的前 9 張圖像，標籤 0 為“女眼”，標籤 1 為“男眼”。

```

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(int(labels[i]))
        plt.axis("off")

```

於深度學習訓練時，需要大量資料以確保訓練時不會產生過度擬合 (over-fitting) 的情況，故使用 Data augmentation 修改、變形資料集中既有的圖片，創造出更多的圖片來讓模型學習。

RandomFlip("horizontal")：水平翻轉

RandomRotation(0.1)：旋轉 0.1 比例

```

data_augmentation = keras.Sequential(
    [
        layers.experimental.preprocessing.RandomFlip("horizontal"),
        layers.experimental.preprocessing.RandomRotation(0.1),
    ]
)

```

展示出增強後之照片，可看出照片已經過翻轉及旋轉。

```
plt.figure(figsize=(10, 10))
for images, _ in train_ds.take(1):
    for i in range(9):
        augmented_images = data_augmentation(images)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
        plt.axis("off")
```



### 3.3 模型建立

前面已設定圖片尺寸為 256 X 256，但是圖片為 RGB 色彩，其值的範圍為 [0,255]，因此首先利用 Rescaling 將值輸入重新縮放到 [0,1] 範圍內。

卷積層：使用二維 (Conv2D) 卷積層以擷取圖片特徵，卷積層共四層。其中 32 代表找出 32 個特徵 (Feature Map)，strides = 2 代表每隔兩步就掃描、padding = "same" 代表掃出來的特徵跟原本的照片比較，要一樣大、每層中的激活函數為使用 Relu。

BatchNormalization 為將輸出標準化，可增加準確度。

擷取完特徵值後利用一層池化層 (Pooling Layer) 將特徵壓縮，使用二維 (MaxPooling2D) 取最大值 (Max)



```

x = layers.experimental.preprocessing.Rescaling(1.0 / 255)(x)
x = layers.Conv2D(32, 3, strides=2, padding="same")(x)
x = layers.BatchNormalization()(x)
x = layers.Activation("relu")(x)

x = layers.Conv2D(64, 3, padding="same")(x)
x = layers.BatchNormalization()(x)
x = layers.Activation("relu")(x)

previous_block_activation = x # Set aside residual

for size in [128, 256, 512, 728]:
    x = layers.Activation("relu")(x)
    x = layers.SeparableConv2D(size, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)

    x = layers.Activation("relu")(x)
    x = layers.SeparableConv2D(size, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)

    x = layers.MaxPooling2D(3, strides=2, padding="same")(x)

```

Dropout(0.5)為了防止過度擬合 (over-fitting)，訓練過程中隨機丟掉一部分，0.5 代表隨機丟掉 50%的輸出單元。

最後，Dense 輸出層

```

x = layers.Dropout(0.5)(x)
outputs = layers.Dense(units, activation=activation)(x)

```

### 3.5 訓練模型

ModelCheckpoint：由於訓練過程耗時，有可能訓練一半就當掉，因此，利用 Callback，在每一個檢查點 (Checkpoint) 存檔，下次執行時，就可以從中斷點繼續訓練。

本研究使用之優化器 (Optimizer) 為 Adam，損失函數為 binary\_crossentropy，起始學習率為 1.0，衰減為 1e-3。

總共訓練 10 次，初始之 accuracy 並不佳，後面 accuracy 逐漸提升至 90%以上。

```

callbacks = [ keras.callbacks.ModelCheckpoint("save_at_{epoch}.h5")]
model.compile(optimizer=keras.optimizers.Adam(1e-3), loss="binary_crossentropy", metrics=["accuracy"])
model.fit(train_ds, epochs=epochs, callbacks=callbacks, validation_data=val_ds)

Epoch 1/10
185/185 [=====] - 3172s 17s/step - loss: 0.6426 - accuracy: 0.6517 - val_loss: 0.7123 - val_accuracy: 0.5553
Epoch 2/10
185/185 [=====] - 3169s 17s/step - loss: 0.3446 - accuracy: 0.8506 - val_loss: 0.7411 - val_accuracy: 0.5553
Epoch 3/10
185/185 [=====] - 3116s 17s/step - loss: 0.2926 - accuracy: 0.8757 - val_loss: 0.2514 - val_accuracy: 0.8889
Epoch 4/10
185/185 [=====] - 3137s 17s/step - loss: 0.2681 - accuracy: 0.8877 - val_loss: 0.6227 - val_accuracy: 0.7731
Epoch 5/10
185/185 [=====] - 3198s 17s/step - loss: 0.2393 - accuracy: 0.9002 - val_loss: 0.2656 - val_accuracy: 0.8972
Epoch 6/10
185/185 [=====] - 3180s 17s/step - loss: 0.2212 - accuracy: 0.9076 - val_loss: 0.2335 - val_accuracy: 0.8928
Epoch 7/10
185/185 [=====] - 3242s 18s/step - loss: 0.2028 - accuracy: 0.9172 - val_loss: 0.2772 - val_accuracy: 0.8911
Epoch 8/10
185/185 [=====] - 3287s 18s/step - loss: 0.1894 - accuracy: 0.9256 - val_loss: 0.9000 - val_accuracy: 0.7184
Epoch 9/10
185/185 [=====] - 3343s 18s/step - loss: 0.1727 - accuracy: 0.9309 - val_loss: 0.1726 - val_accuracy: 0.9302
Epoch 10/10
185/185 [=====] - 3245s 18s/step - loss: 0.1756 - accuracy: 0.9272 - val_loss: 0.2203 - val_accuracy: 0.9158

```

最後，使用此模型對未包含在訓練集或驗證集中的圖像進行辨識。

```

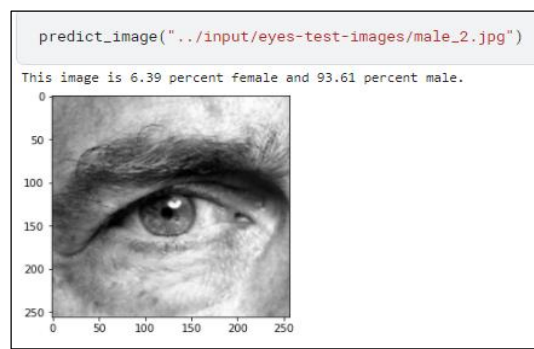
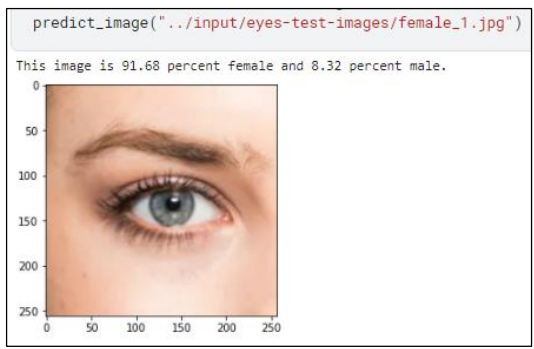
def predict_image(image_path):
    img = keras.preprocessing.image.load_img(image_path, target_size=image_size)
    img_array = keras.preprocessing.image.img_to_array(img)

    img_array = tf.expand_dims(img_array, 0)

    predictions = model.predict(img_array)
    score = predictions[0]
    print("This image is %.2f percent female and %.2f percent male." % (100 * (1 - score), 100 * score))
    plt.imshow(img);

```

任取外部的圖片以驗證模型之泛化能力。



## 第四章、結論與未來展望

本研究利用 CNN 方法以人眼辨識出性別，最後利用資料集外的圖片以驗證模型訓練的結果，準確率可達 90% 以上。本研究欲提供 CNN 模型予警方或相關偵查單位，以提高辦案之效率。然而，目前此模型僅能以人眼辨識性別，有時可能較難取得犯罪者眼睛的照片，或者取得的犯罪者眼睛照片為不清楚的，將大大影響模型之成效。因此，我認為未來若能增加更多身體部位的照片加以訓練並建構模型，便能以更多方式辨識出性別。此外，此模型可結合各監視器，使警方、偵查單位等得到即時之資訊，更加快速地達成任務。

## 第五章、參考文獻

- Ketkar, N., & Santana, E. (2017). Deep Learning with Python (Vol. 1): Springer. 15
- <https://www.kaggle.com/pavelbiz/eyes-rtte>
- [https://keras.io/examples/vision/image\\_classification\\_from\\_scratch/](https://keras.io/examples/vision/image_classification_from_scratch/)
- <https://www.tensorflow.org/tutorials/images/classification?hl=zh-tw>