以深度學習為基礎建構人眼辨識性別模型

110034555 李勁緯
指導教授：邱銘傳 博士

目錄

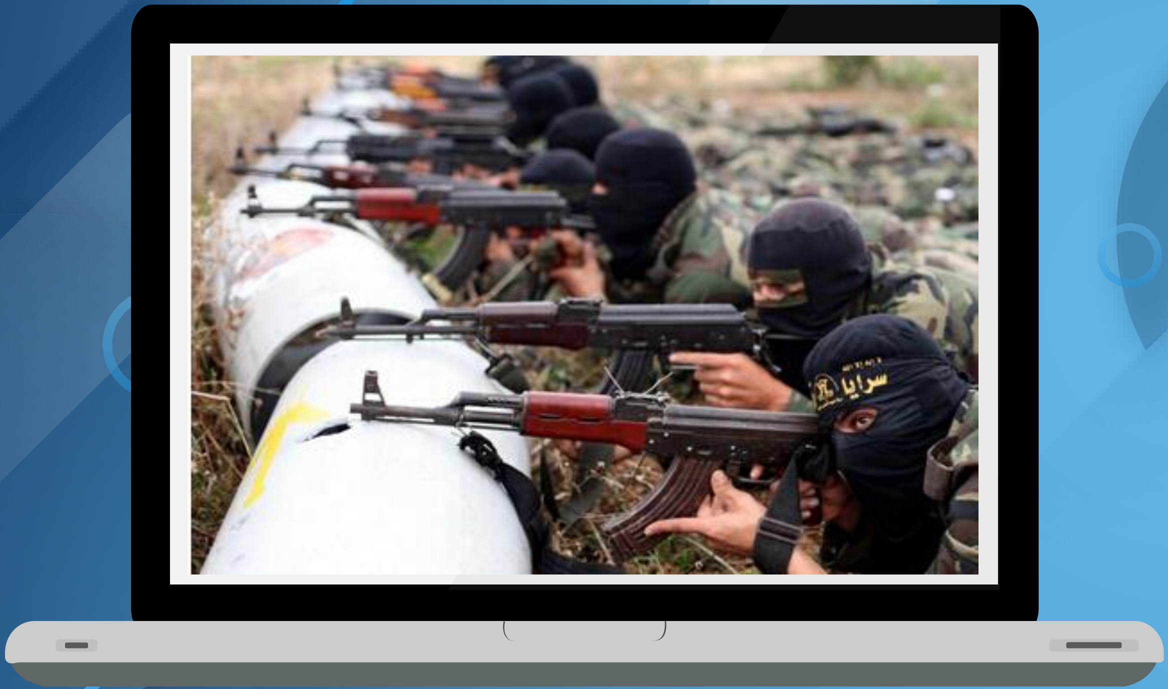
1

研究背景

研究背景

- 金融風暴、疫情肆虐
- 貧富差距擴大
- 竊盜、搶劫等案件層出不窮

研究背景



本研究利用CNN建構人眼辨識性別模型，協助警方藉由人眼照片辨識犯罪者性別，能大幅減少警方偵查的範圍，並快速的將犯罪者繩之以法。

5W1H

警方、相關偵查單位

世界各地

以CNN深度學習之方法建立人眼辨識性別模型

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

蒐集到犯罪者的眼睛照片時

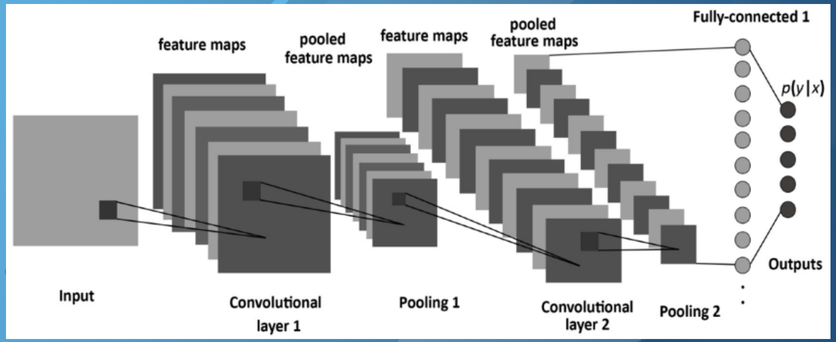
協助警方辦案

2

研究過程

卷積神經網路（CNN）

- CNN的基本結構包含多個交替堆疊的卷積層、池化層級全連階層所組成。
- 可藉由擷取圖片或影像形狀相關的特徵進行圖片或影像辨識。
- 參數數量較少、訓練難度較低，成為一相當具吸引力的深度學習結構。



資料來源

由Kaggle公開數據集中取得人眼的圖像資料集，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。

資料前處理

設定參數，並載入資料集。

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

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

50%

75%

資料前處理

將數據拆分為訓練集和驗證集，比例為8：2

```
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.
```


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

資料擴增

為避免產生過度擬合（over-fitting）的情況，利用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。

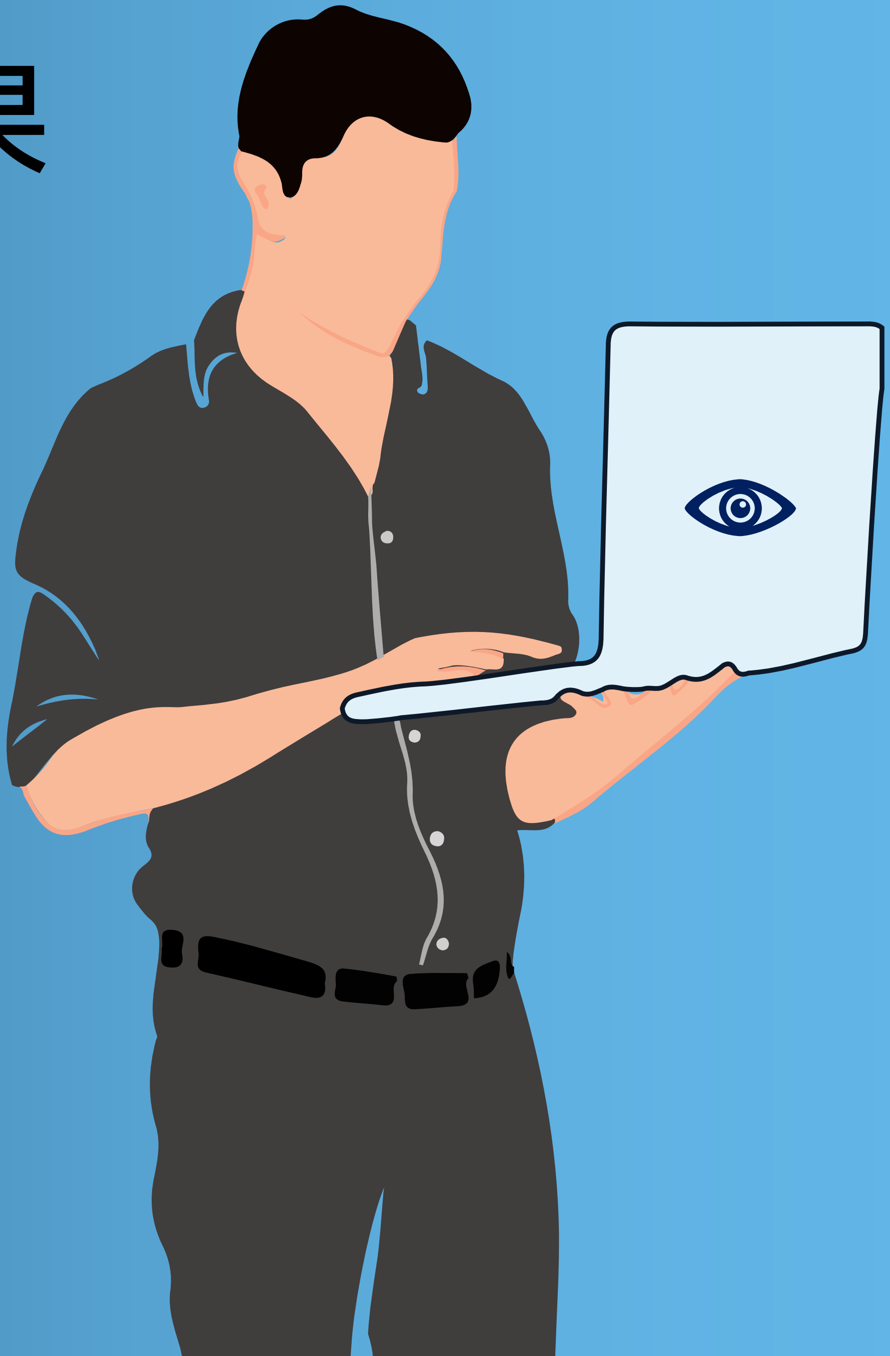
```
data_augmentation = keras.Sequential(
    [
        layers.experimental.preprocessing.RandomFlip("horizontal"),
        layers.experimental.preprocessing.RandomRotation(0.1),
    ]
)
```

水平翻轉

旋轉0.1比例

顯示資料擴增後之結果

```python
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")
```



模型建立

```python
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)
```

- Rescaling將值重新縮放到〔0,1〕範圍內。
- 二維（Conv2D）卷積層，共四層。
- 一層二維（MaxPooling2D）池化層，取特徵最大值。

模型訓練

- 訓練時間長，利用 Callback，在每一個檢查點（Checkpoint）存檔。
- 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。

```
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
```

3

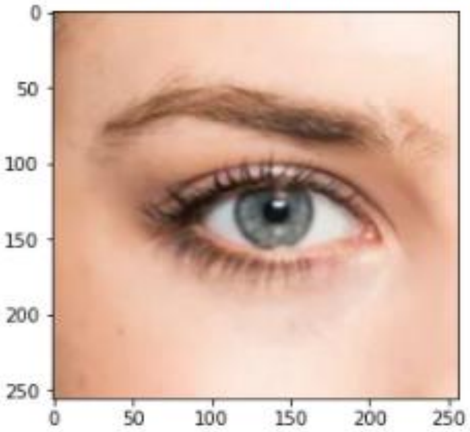
研究結果

模型驗證

任取外部的圖片以驗證模型之泛化能力。

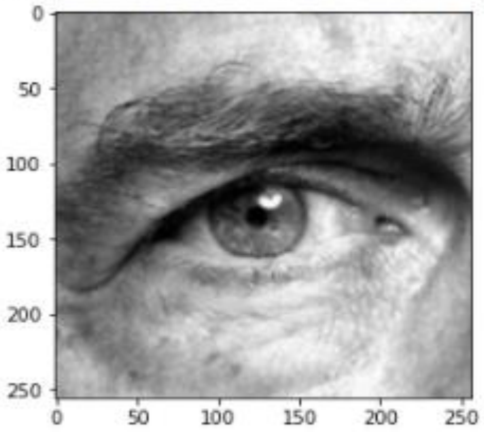
```
predict_image("../input/eyes-test-images/female_1.jpg")
```

This image is 91.68 percent female and 8.32 percent male.




```
predict_image("../input/eyes-test-images/male_2.jpg")
```

This image is 6.39 percent female and 93.61 percent male.



結論與未來展望

- 本研究利用CNN方法以人眼辨識性別，準確率可達90%以上，此模型可提供警方或相關偵查單位，以提高辦案之效率。

- 增加更多身體部位的照片。

- 結合各監視器，使警方、偵查單位等得到即時之資訊。

參考資料

- https://www.kaggle.com/pavelbiz/eyes-rtte
- https://keras.io/examples/vision/image_classification_from_scratch/
- https://www.tensorflow.org/tutorials/images/classification?hl=zh-tw

THANK YOU