

Emotions Recognition

108321013

108328017

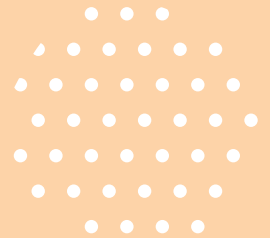
109321011

黃子珊

林育萱

劉昱萬





目錄

01 工作分配

02 資料前處理

03 Model 架構

04 Model 評估



01

工作分配

工作分配

	子珊	育萱	昱萬
資料蒐集	✓	✓	✓
圖像前處理	✓	✓	✓
model訓練		✓	
ppt	✓	✓	

02

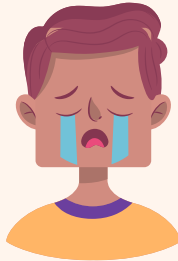
資料前處理



Classification



Happy



SAD



Angry



Normal

PAIR THE CONCEPTS

Sad

Normal

Happy

Angry



Train

160

185

153

152

Valid

40

46

38

37

Test

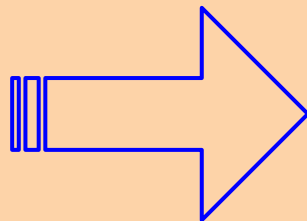
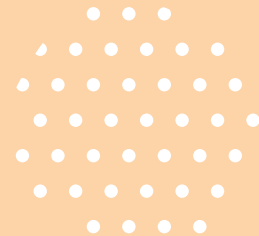
60

60

60

60

Normalize



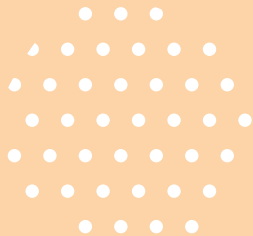
擷取人臉，避免背景複雜影響訓練



03

模型架構

Model choose



經過測試後，InceptionResNetV2表現最好，因此選用此模型

VGG16

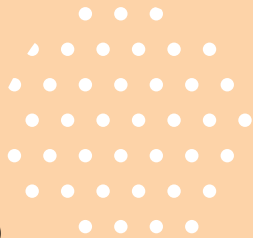
ResNet50

ResNet101

InceptionResNetV2

- Step 1: 設置模型架構
- Step 2: 設置callbacks
- Step 3: 設置資料集
- Step 4: 訓練模型、儲存模型、紀錄學習歷程

Model structure



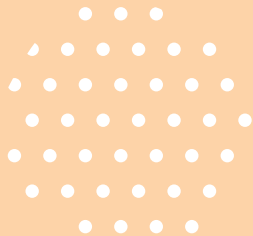
- 使用transfer learning, 引入 InceptionResNetV2 後添加 GlobalAveragePooling2D、Dense、Dropout, optimizer使用Adam

```
# 載入keras模型(更換輸出圖片尺寸)
model = InceptionResNetV2(include_top=False, weights='imagenet', input_tensor=Input(shape=(80, 80, 3)))

# # 定義輸出層
x = model.output
x = GlobalAveragePooling2D()(x)
x = Dense(64, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(4, activation='softmax')(x) → 最後分成4個類別
model = Model(inputs=model.input, outputs=predictions)

# 編譯模型
model.compile(optimizer=Adam(lr=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Callbacks



- 設置earllystop (當val_loss 超過10 epoch沒有更新權重即停止訓練)、checkpoint、learning rate降低條件

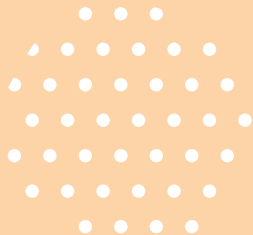
```
# 設定earllystop條件
estop = EarlyStopping(monitor='val_loss', patience=10, mode='min', verbose=1)

# 設定模型儲存條件
checkpoint = ModelCheckpoint('InceptionResNetV2_checkpoint_v2.h5', verbose=1,
                             monitor='val_loss', save_best_only=True,
                             mode='min')

# 設定lr降低條件
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
                               patience=5, mode='min', verbose=1,
                               min_lr=0.0001)
```

- [illegible]

Train



- Epoch設為50, 由於有使用 earlystop, 當權重一直沒更新, 即會停止訓練
- Batch size: 64

```
# -----4.訓練模型-----  
# 重新訓練權重  
history = model.fit_generator(train_generator,  
                             epochs = 50, verbose = 1,  
                             steps_per_epoch = train_generator.samples//batch_size,  
                             validation_data = valid_generator,  
                             validation_steps = valid_generator.samples//batch_size,  
                             callbacks=[checkpoint, estop, reduce_lr])  
  
# -----5.儲存模型、紀錄學習歷程-----  
# 儲存模型  
model.save('./InceptionResNetV2_retrained_v2.h5')  
print('已儲存InceptionResNetV2_retrained_v2.h5')  
  
show_train_history(history)
```

04

模型評估



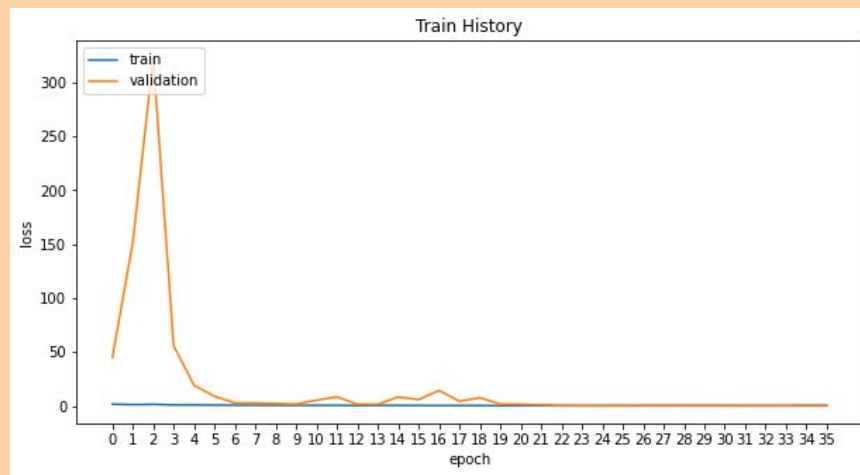
Train history

Accuracy



Train accuracy: 0.973
Valid accuracy: 0.875

Loss

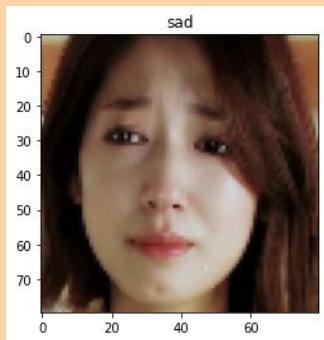
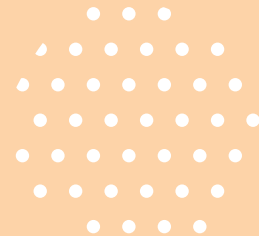


Train loss: 1.24
Valid loss: 0.476

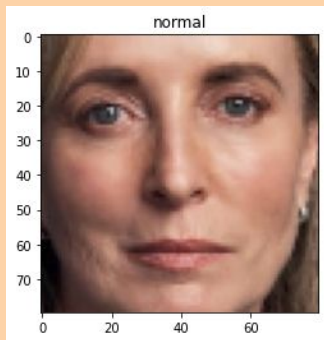
Test

Test accuracy: 0.8125

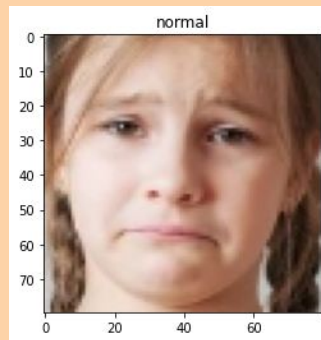
Test loss: 1.083



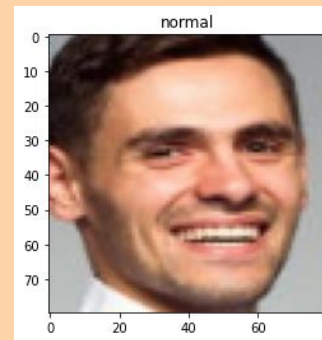
Output: sad



Output: normal



Output: normal



Output: normal