

目錄



○ 工作分配

02 資料前處理

O3 Model 架構

04 Model 評估



01 工作分配

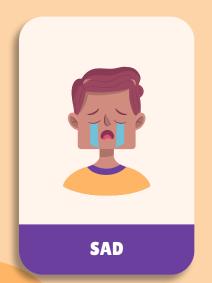
工作分配

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

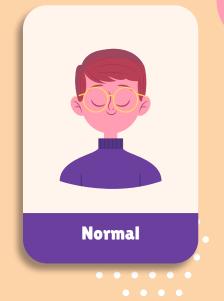


Classification





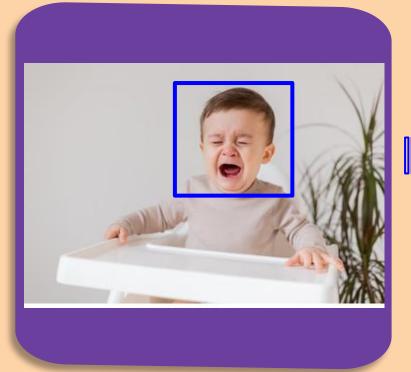


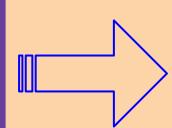


PAIR THE CONCEPTS

	Sad	Normal	Нарру	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)
                                                 → 最後分成4個類別
predictions = Dense(4, activation='softmax')(x)
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earlystop (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earlystop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)
```

Data Augmentation

- 使用ImageDataGenerator做資料擴增
- 對Training data做正規化、擴增
- valid、test data只做正規化,不做擴增
- Batch size: 64
- Target size: (80,80)

```
# 設定ImageDataGenerator參數(路徑、批量、圖片尺寸)
train dir = './train/'
valid dir = './valid/'
test dir = './test/'
batch size = 64
target size = (80, 80)
##設定批量生成器
train datagen = ImageDataGenerator(rescale=1./255,
                                   rotation range=20,
                                  width shift range=0.2,
                                   height shift range=0.2,
                                   shear range=0.2,
                                   fill mode="nearest")
val datagen = ImageDataGenerator(rescale=1./255)
test datagen = ImageDataGenerator(rescale=1./255)
##讀取資料集+批量生成器,產生每epoch訓練樣本
train generator = train datagen.flow from directory(train dir,
                                     target size=target size,
                                     batch size=batch size)
valid generator = val datagen.flow from directory(valid dir,
                                     target size=target size,
                                     batch size=batch size)
test generator = test datagen.flow from directory(test dir,
                                     target size=target size,
                                     batch size=batch size,
                                     shuffle=False)
```

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



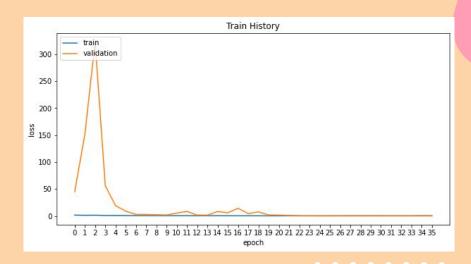
Train history

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





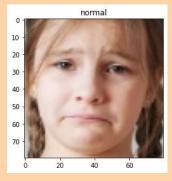
Output: sad

70 -

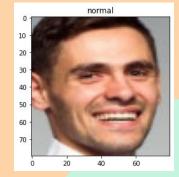


Output: normal





Output: normal



Output: normal