國立清華大學
NATIONAL TSING HUA UNIVERSITY

工業工程與
工程管理學系

智慧化企業整合

Pneumonia

Using CNN Model to Classify Chest X-Ray

人工智慧醫療診斷模型

學號：108034401
姓名：蘇靜琦

國立清華大學
NATIONAL TSING HUA UNIVERSITY

目錄

一. 研究背景
二. 問題定義
三. 研究方法
四. 研究結果
五. 結論
六. 參考資料

國立清華大學
NATIONAL TSING HUA UNIVERSITY

研究背景

- 肺炎是現代非常常見的疾病
- 2013年台灣有9042人死於肺炎，平均<1小時有1人因肺炎喪生
- 10 年間成長了兩倍，已在 2016 年成為台灣十大死因的第三名
- 初期症狀像感冒，很常被忽略、感染肺炎的年齡層不拘
- 短至一天、甚至幾小時內，病菌就可能快速大量繁殖
- 台灣 X 光專科醫師人數有限
- 利用AI挑出可能罹患肺炎的 X 光影像，再交由 X 光科醫師去判讀，才是兼顧效率與醫療價值的做法
- 提高辨認肺炎病徵的準確率，縮短細菌的鑑定時間，以爭取更多的時間搶救病患

國立清華大學
NATIONAL TSING HUA UNIVERSITY

問題定義-5W1H

What　單獨由醫生判斷錯誤，可能準確率較低的問題

When　醫生確認並辨別病人肺炎的時候

Who　醫院看診醫生採用此模型，增加判斷準確率

Where　醫院的胸腔內科

Why　避免錯誤診斷導致病人看診時間延誤、提高醫院胸腔內科的名聲、給予病人更好的治療

How　建構神經網路模型，協助醫生進行病人肺炎病徵的判斷

國立清華大學
NATIONAL TSING HUA UNIVERSITY

研究方法

建構人工智慧醫療診斷模型的過程大致可以分為七個步驟：

- 資料收集、分類
- 資料前處理
- 建立神經網路架構
- 訓練神經網路模型
- 透過測試集預測神經網路成效
- 不斷評估模型，優化此模型
- 提出結果

研究方法

資料收集、分類

- 網路上開源的資料集，共有2 種不同病徵的肺炎照片 ，共 5856 張。
- 其中包括正常的肺部胸腔X光片1583張和罹患肺炎的肺部胸腔X光片4273張，將其作為模型的投入。

研究結果

AS-IS原始模型當中，CNN模型的架構為:

- 照片輸入(150，150，3)
- 一層卷積核大小為 3x3，個數為 32 的卷積層
- 一層池化大小為 2x2 的池化層
- 進行比率為 0.4 的 Dropout
- 二層卷積核大小為 3x3，個數為 32 的卷積層
- 一層池化大小為 2x2 的池化層
- 進行比率為 0.25的 Dropout
- 三層卷積核大小為 3x3，個數為 64 的卷積層
- 一層池化大小為 2x2 的池化層
- 一層隱藏單元為 64 個的全連接層
- 進行比率為 0.5 的 Dropout
- 各層激活函數:relu
- 輸出層激活函數:softmax
- 損失函數: categorical_crossentrophy
- 優化器：Rmsprop

國立清華大學
NATIONAL TSING HUA UNIVERSITY

研究結果

AS-IS原始模型

- 訓練集的準確率為 0.7957，驗證集準確率為 0.6875，測試集準確率為 0.7708
- 病徵分類的準確率仍偏低

國立清華大學
NATIONAL TSING HUA UNIVERSITY

研究結果

TO-BE最終模型架構如下:

- 照片輸入(150，150，3)
- 一層卷積核大小為 3x3，個數為 32 的卷積層
- 一層池化大小為 2x2 的池化層
- 進行比率為 0.4 的 Dropout
- 二層卷積核大小為 3x3，個數為 32 的卷積層
- 一層池化大小為 2x2 的池化層
- 三層卷積核大小為 3x3，個數為 64 的卷積層
- 一層池化大小為 2x2 的池化層
- 四層卷積核大小為 3x3，個數為 64 的卷積層
- 一層池化大小為 2x2 的池化層
- 一層隱藏單元為 128 個的全連接層
- 進行比率為 0.5 的 Dropout

- 各層激活函數: relu
- 輸出層激活函數: sigmoid
- 損失函數: binary_crossentrophy
- 優化器：Adam
- Epochs：50
- Batchsizes：16

研究方法

資料前處理

原始模型并沒進行資料的前處理，改善後模型的前處理可分為下面幾個步驟:

i. 將放置在各個不同的資料夾的圖片進行讀取

ii. 將所有圖片裁減為相同大小，使每張的像素點數一致，才能夠放入模型中

iii. 將區分資料集的隨機性固定

iv. 將資料集的 80%做為訓練集，20%做為測試集。訓練集當中再取出 10%做為驗證集

v. 將圖片的像素值除以255(將像素值介於 0-255 之間)，使像素值介於 0-1 之間

vi. 在訓練過程，將圖片隨機剪切 20 度，隨機放大縮小 20%，圖片水平隨机翻轉，加強模型的訓練。

研究方法

資料前處理

```
#我們將圖像調整為150 * 150
img_width, img_height = 150, 150
#讀取所有的資料集
train_data_dir = 'C:/Users/Sandy/Desktop/chest_xray/chest_xray/train'
validation_data_dir = 'C:/Users/Sandy/Desktop/chest_xray/chest_xray/val'
test_data_dir = 'C:/Users/Sandy/Desktop/chest_xray/chest_xray/test'
#返回默認的圖像維度順序
if K.image_data_format() == 'channels_first':
    input_shape = (3, img_width, img_height)
else:
    input_shape = (img_width, img_height, 3)
```

Users: str

研究方法

資料前處理

```
#圖片生成器
# this is the augmentation configuration we will use for training
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

#只有 rescaling
test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')
```

研究結果

```
# coding=utf8
#加入我們所需要使用的套件
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.layers import Activation, Dropout, Flatten, Dense, Conv2D, MaxPool2D
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras import backend as K
import numpy as np
import tensorflow as tf
from sklearn import svm
import time
import os
import pandas as np
import matplotlib.pyplot as plt
```

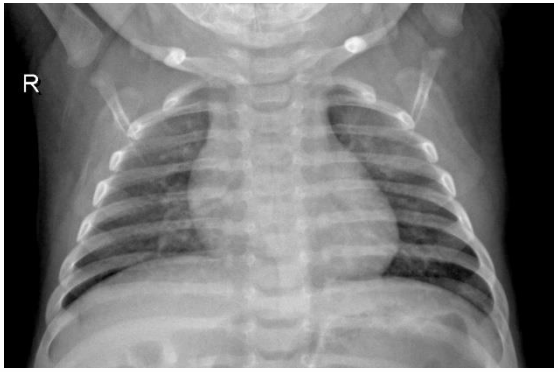
國立清華大學
NATIONAL TSING HUA UNIVERSITY

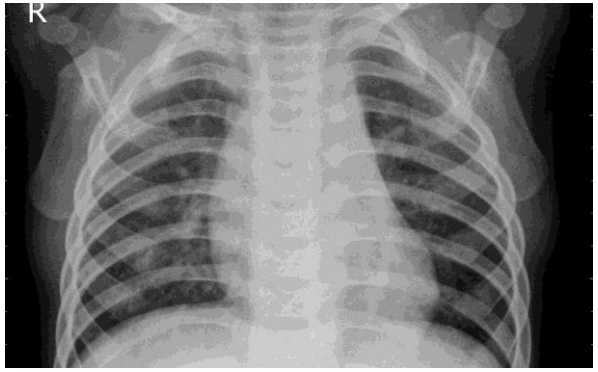
研究結果

```
#顯示出目前的資料、Img
print(os.listdir("C:/Users/Sandy/Desktop/chest_xray/chest_xray"))
print(os.listdir("C:/Users/Sandy/Desktop/chest_xray/chest_xray/train"))
print(os.listdir("C:/Users/Sandy/Desktop/chest_xray/chest_xray/test"))

img_name = 'NORMAL2-IM-0588-0001.jpeg'
img_normal = load_img('C:/Users/Sandy/Desktop/chest_xray/chest_xray/train/NORMAL/' + img_name)
print('NORMAL')
plt.imshow(img_normal)
plt.show()

img_name = 'person63_bacteria_306.jpeg'
img_pneumonia = load_img('C:/Users/Sandy/Desktop/chest_xray/chest_xray/train/PNEUMONIA/' + img_name)
print('PNEUMONIA')
plt.imshow(img_pneumonia)
plt.show()
```





研究結果

```
#我們將圖像調整為150 * 150
img_width, img_height = 150, 150
#讀取所有的資料集
train_data_dir = 'C:/Users/Sandy/Desktop/chest_xray/chest_xray/train'
validation_data_dir = 'C:/Users/Sandy/Desktop/chest_xray/chest_xray/val'
test_data_dir = 'C:/Users/Sandy/Desktop/chest_xray/chest_xray/test'
#設定參數
nb_train_samples = 5216
nb_validation_samples = 624
epochs = 100
batch_size = 16
#返回默認的圖像維度順序
if K.image_data_format() == 'channels_first':
    input_shape = (3, img_width, img_height)
else:
    input_shape = (img_width, img_height, 3)
```

研究結果

```
#建立CNN架構
#照片輸入(150，150，3)
#一層卷積核大小為 3x3，個數為 32 的卷積層
#一層池化大小為 2x2 的池化層
#進行比率為 0.4 的 Dropout
#二層卷積核大小為 3x3，個數為 32 的卷積層
#一層池化大小為 2x2 的池化層
#三層卷積核大小為 3x3，個數為 64 的卷積層
#一層池化大小為 2x2 的池化層
#四層卷積核大小為 3x3，個數為 64 的卷積層
#一層池化大小為 2x2 的池化層
#一層隱藏單元為 128 個的全連接層
#進行比率為 0.5 的 Dropout
#各層激活函數: relu
#輸出層激活函數: sigmoid
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', padding = 'Same', input_shape=input_shape))
model.add(MaxPool2D(pool_size = (2, 2)))
model.add(Dropout(0.4))

model.add(Conv2D(32, (3, 3), activation='relu',padding = 'Same'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu',padding = 'Same'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu',padding = 'Same'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
```

研究結果

```
#決定模型的優化器，損失函數
#用adam的優化器優化目標函式
#loss設定二項交叉熵的形式
model.compile(optimizer = 'adam' , loss = "binary_crossentropy", metrics=["accuracy"])

#圖片生成器
# this is the augmentation configuration we will use for training
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

#只有 rescaling
test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')
```

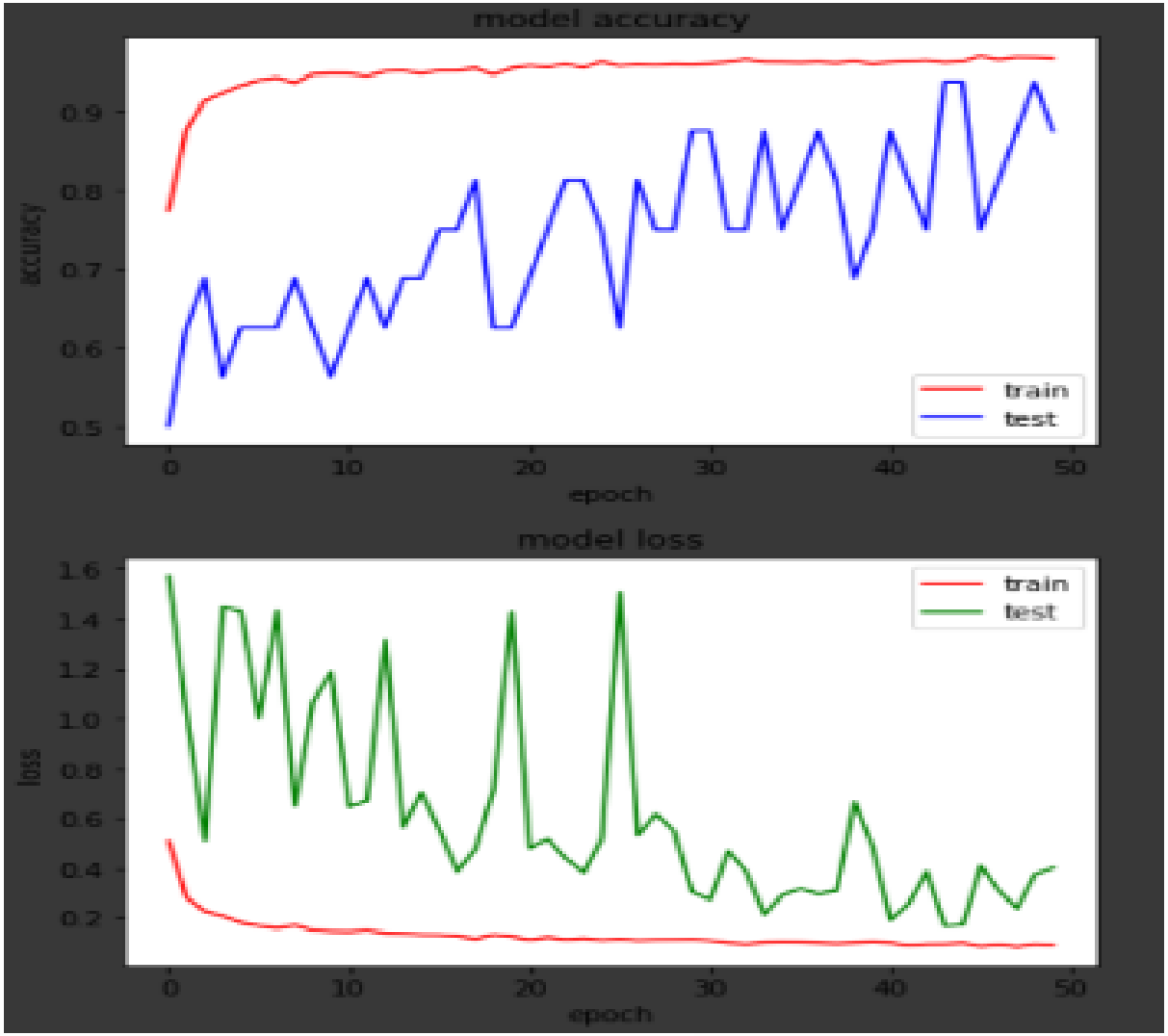
研究結果

```
#訓練模型
history = model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples//batch_size, #steps_per_epoch stand for number of batchs
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples//batch_size)

#存取模型權重為h5檔
model.save_weights('first_try.h5')
#評估模型
scores = model.evaluate_generator(test_generator)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
#畫出Accuracy、Loss的圖
fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_facecolor('w')
ax.grid(b=False)
ax.plot(history.history['acc'], color='red')
ax.plot(history.history['val_acc'], color ='blue')
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='lower right')
plt.show()
fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_facecolor('w')
ax.grid(b=False)
ax.plot(history.history['loss'], color='red')
ax.plot(history.history['val_loss'], color ='green')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper right')
plt.show()
```

國立清華大學
NATIONAL TSING HUA UNIVERSITY

研究結果



國立清華大學 NATIONAL TSING HUA UNIVERSITY

研究結果

損失函數、激活函數

AS-IS

- Softmax
- Categorical_crossentrop

TO-BE

- Sigmoid
- Binary_crossentropy

激活函數	損失函數	優化器	訓練集準確率	驗證集準確率	測試集準確率
softmax	Categorical crossentropy	Adam	0.9689	0.8125	0.8622
softmax	Binary crossentropy	Adam	0.9703	0.6875	0.7708
sigmoid	Binary crossentropy	Adam	0.9758	0.9375	0.9311
sigmoid	Categorical crossentropy	Adam	0.9699	0.8125	0.9135

研究結果

決定優化器

AS-IS

- Rmsprop

TO-BE

- Adam

激活函數	損失函數	優化器	訓練集準確率	驗證集準確率	測試集準確率
sigmoid	Binary crossentropy	Adam	0.9758	0.9375	0.9311
sigmoid	Binary crossentropy	Rmsprop	0.9534	0.8750	0.8638

研究結果

其他調整

AS-IS

- 無Generator
- 多層Dropout

TO-BE

- Generator
- 減少Dropout

Generator	Dropout	優化器	訓練集準確率	驗證集準確率	測試集準確率
-	0.4/0.25/0.5	Adam	0.9438	0.7500	0.8349
V	0.5	Adam	0.9758	0.9375	0.9311

研究總結-結論

結論

- 訓練集準確率達 0.9758、測試集準確率達0.9311
- 比原模型訓練集準確率 0.7957、測試集 0.7708 有著非常顯著的提升，可見這次針對模型的改善，的確有使模型的預測能力提升
- 93.11%的準確率可協助醫生在判斷病人肺炎病徵時，能夠協助醫生做出正確的判斷，讓病人能夠接受正確的治療

研究總結-研究限制

研究限制

- 模型準確率無法提高到 94%以上
- 主要受到下面幾點的原因所影響:

a) 2種X光片的張數較少且不平均

b) 訓練一次(50 個 Epoch)需要 2-3小時

研究總結-未來研究方向

未來研究方向

- 增加圖片張數較少的X光片數量
- 針對醫學影像分類特點

利用兩種結構、深度不同的神經網路模型AlexNet 與 Inception V3

以上兩種優化後的模型 AlexNet_S：

a) 在分類性能上接近甚至超過 Inception V3 模型

b) 大幅降低了模型對系統資源的需求

在未來可進一步提高模型的性能，同時減少模型對系統資源的需求。

參考資料

- XU Shan-shan，LIU Ying-an*，XU Sheng. Wood defects recognition based on the convolutional neural network. Journal of SHANDONG University (ENGINEERING SCIENCE) ，2013，43(2): 23-28.
- Ciresan DC.，Meier U，Masci J，et al． Flexible，high performance convolutional neural networks for image classification. Proceedings of the 22nd International Joint Conference on Artificial Intelligence，Volume 2.AIPress，2011:1237-1242．

國立清華大學
NATIONAL TSING HUA UNIVERSITY

Thanks For Your Attention