**無線網路，第一次作業HWK-002-題目: 即時車牌影像辨識**

**四資工三甲\_C110151138\_黃吉菘\_20240616**

* *Topic*

**這個專案旨在實現一個車牌辨識系統，該系統能夠從圖像中檢測車牌，並使用預訓練模型來識別車牌上的文字。**

* *程式碼:*

resize.py

def emptydir(dirname):  #清空資料夾

    if os.path.isdir(dirname):  #資料夾存在就刪除

        shutil.rmtree(dirname)

        sleep(2)  #需延遲,否則會出錯

    os.mkdir(dirname)  #建立資料夾

def dirResize(src, dst):

    myfiles = glob.glob(src + '/\*.JPG')  #讀取資料夾全部jpg檔案

    emptydir(dst)

    print(src + ' 資料夾：')

    print('開始轉換圖形尺寸！')

    for f in myfiles:

        fname = f.split("\\")[-1]

        img = Image.open(f)

        img\_new = img.resize((300, 225), PIL.Image.ANTIALIAS)  #尺寸300x225

        img\_new.save(dst + '/' + fname)

    print('轉換圖形尺寸完成！\n')

import PIL

from PIL import Image

import glob

import shutil, os

from time import sleep

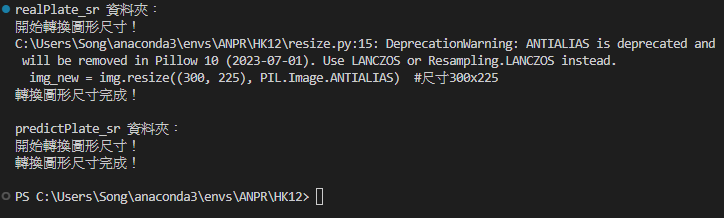
dirResize('realPlate\_sr', 'realPlate')

dirResize('predictPlate\_sr', 'predictPlate')

* *程式執行及說明:*

此程式首先定義了一個emptydir函數，用於清空並重新建立指定的資料夾。接著定義了一個dirResize函數，用於讀取來源資料夾中的所有JPG圖片，將它們的尺寸轉換為300x225，然後保存到目標資料夾中。在主程式中，調用dirResize函數對兩個資料夾進行處理：realPlate\_sr和predictPlate\_sr。

* *程式輸出與結果描述:*



* *問題討論與程式修正:*

無

* *程式碼:*

cropPlate.py

def emptydir(dirname):  #清空資料夾

    if os.path.isdir(dirname):  #資料夾存在就刪除

        shutil.rmtree(dirname)

        sleep(2)  #需延遲,否則會出錯

    os.mkdir(dirname)  #建立資料夾

import cv2

from PIL import Image

import glob

import shutil, os

from time import sleep

print('開始擷取車牌！')

print('無法擷取車牌的圖片：')

dstdir = 'cropPlate'

myfiles = glob.glob("realPlate\\*.JPG")

emptydir(dstdir)

for imgname in myfiles:

    filename = (imgname.split('\\'))[-1]  #取得檔案名稱

    img = cv2.imread(imgname)  #讀入圖形

    detector = cv2.CascadeClassifier('haar\_carplate.xml')

    signs = detector.detectMultiScale(img, scaleFactor=1.1, minNeighbors=4, minSize=(20, 20))  #框出車牌

    #割取車牌

    if len(signs) > 0 :

        for (x, y, w, h) in signs:

            image1 = Image.open(imgname)

            image2 = image1.crop((x, y, x+w, y+h))  #擷取車牌圖形

            image3 = image2.resize((140, 40), Image.ANTIALIAS)  #轉換尺寸為140X40

            image3.save(dstdir + '/tem.jpg')

            image4 = cv2.imread(dstdir + '/tem.jpg')  #以opencv讀車牌檔

            img\_gray = cv2.cvtColor(image4, cv2.COLOR\_RGB2GRAY)  #灰階

            \_, img\_thre = cv2.threshold(img\_gray, 100, 255, cv2.THRESH\_BINARY)  #黑白

            cv2.imwrite(dstdir + '/'+ filename, img\_thre)

    else:

        print(filename)

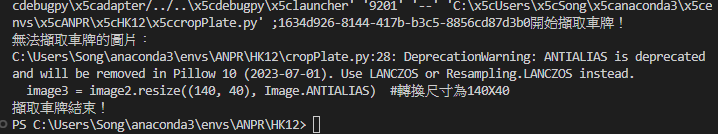
os.remove(dstdir + '/tem.jpg')  #移除暫存檔

print('擷取車牌結束！')

* *程式執行及說明:*

此程式首先定義了一個emptydir函數，用於清空並重新建立指定的資料夾。接著，在主程式中，讀取來源資料夾realPlate中的所有JPG圖片，利用Haar特徵級聯分類器檢測車牌位置，將檢測到的車牌部分擷取出來，並轉換為140x40的尺寸。然後對擷取的車牌圖像進行灰階和二值化處理，最終將處理後的車牌圖像保存到目標資料夾cropPlate中。如果未能檢測到車牌，則輸出對應的圖片名稱。

* *程式輸出與結果描述:*



* *問題討論與程式修正:*

無

* *程式碼:*

cropNum\_all.py

def emptydir(dirname):  #清空資料夾

    if os.path.isdir(dirname):  #資料夾存在就刪除

        shutil.rmtree(dirname)

        sleep(2)  #需延遲,否則會出錯

    os.mkdir(dirname)  #建立資料夾

import cv2

import glob

import shutil, os

from time import sleep

print('開始擷取車牌數字！')

emptydir('cropNum')

myfiles = glob.glob('cropPlate\\*.jpg')

for f in myfiles:

    filename = (f.split('\\'))[-1].replace('.jpg', '')  #移除檔名中的「.jpg」

    emptydir('cropNum/' + filename)  #以車牌號碼做資料夾名稱

    image = cv2.imread(f)  #讀取車牌號碼圖片

    gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  #灰階

    \_,thresh = cv2.threshold(gray, 127, 255, cv2.THRESH\_BINARY\_INV)  #轉為黑白

    contours1 = cv2.findContours(thresh.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  #尋找輪廓

    contours = contours1[1]  #取得輪廓

    letter\_image\_regions = []  #文字圖形串列

    for contour in contours:  #依序處理輪廓

        (x, y, w, h) = cv2.boundingRect(contour)  #單一輪廓資料

        letter\_image\_regions.append((x, y, w, h))  #輪廓資料加入串列

    letter\_image\_regions = sorted(letter\_image\_regions, key=lambda x: x[0])  #按X坐標排序

    #存檔

    i=0

    for letter\_bounding\_box in letter\_image\_regions:  #依序處理輪廓資料

        x, y, w, h = letter\_bounding\_box

        if w>=5 and h>32 and h<40:  #長度>6且高度在30-48才是文字

            letter\_image = gray[y:y+h, x:x+w]  #擷取圖形

            letter\_image = cv2.resize(letter\_image, (18, 38))

            cv2.imwrite('cropNum/' + filename + '/{}.jpg'.format(i+1), letter\_image)  #存各車牌文字檔

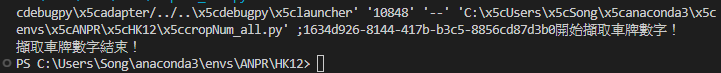
            i += 1

print('擷取車牌數字結束！')

* *程式執行及說明:*

此程式首先定義了一個emptydir函數，用於清空並重新建立指定的資料夾。接著，在主程式中，讀取來源資料夾cropPlate中的所有車牌圖像，將每個車牌圖像轉為灰階並進行二值化處理，然後利用輪廓檢測方法尋找車牌數字的輪廓，並對這些數字進行擷取和尺寸調整，最終將擷取的數字圖像保存到以車牌號碼命名的資料夾中。

* *程式輸出與結果描述:*



* *問題討論與程式修正:*

無

* *程式碼:*

makefont.py

def emptydir(dirname):  #清空資料夾

    if os.path.isdir(dirname):  #資料夾存在就刪除

        shutil.rmtree(dirname)

        sleep(2)  #需延遲,否則會出錯

    os.mkdir(dirname)  #建立資料夾

import glob

import shutil, os

from time import sleep

print('開始建立文字庫！')

emptydir('platefont')

fontlist = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','J','K','L','M','N','P','Q','R','S','T','U','V','W','X','Y','Z']

for i in range(len(fontlist)):  #建立文字資料夾

    emptydir('platefont/' + fontlist[i])

dirs = os.listdir('cropNum')  #讀取所有檔案及資料夾

picnum = 1  #圖片記數器,讓檔名不會重複

for d in dirs:

    if os.path.isdir('cropNum/' + d):  #只處理資料夾

        myfiles = glob.glob('cropNum/' + d + '/\*.jpg')

        for i, f in enumerate(myfiles):

            shutil.copyfile(f, 'platefont/{}/{}.jpg'.format(d[i], picnum))  #存入對應資料夾

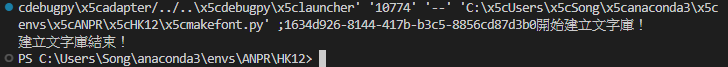
            picnum += 1

print('建立文字庫結束！')

* *程式執行及說明:*

此程式首先定義了一個emptydir函數，用於清空並重新建立指定的資料夾。接著，在主程式中，建立一個存放文字圖像的目標資料夾platefont，並為每個字母和數字建立對應的子資料夾。然後，讀取來源資料夾cropNum中的所有車牌數字資料夾，將這些資料夾中的每個圖像依次複製到目標資料夾中對應的子資料夾，並確保每個圖像的檔名不重複。

* *程式輸出與結果描述:*



* *問題討論與程式修正:*

無

* *程式碼:*

makedata.py

def emptydir(dirname):  #清空資料夾

    if os.path.isdir(dirname):  #資料夾存在就刪除

        shutil.rmtree(dirname)

        sleep(2)  #需延遲,否則會出錯

    os.mkdir(dirname)  #建立資料夾

import cv2

import random

import glob

import shutil, os

from time import sleep

fontlist = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','J','K','L','M','N','P','Q','R','S','T','U','V','W','X','Y','Z']

print('開始建立訓練資料！')

emptydir('data')

for n in range(len(fontlist)):

    print('產生 data/' + fontlist[n] +' 資料夾')

    emptydir('data/' + fontlist[n])

    myfiles = glob.glob('platefont/' + fontlist[n] + '/\*.jpg')

    for index, f in enumerate(myfiles):

        pic\_total = 500  #每個文字檔案數

        pic\_each = int(pic\_total / len(myfiles)) + 1

        for i in range(pic\_each):  #i 為檔案名稱

            img = cv2.imread(f)

            for j in range(20):  #加入指定數量雜點

                x = random.randint(0, 17)  #以亂數設定位置

                y = random.randint(0, 37)

                cv2.circle(img, (x, y), 1, (0,0,0), -1)  #畫點

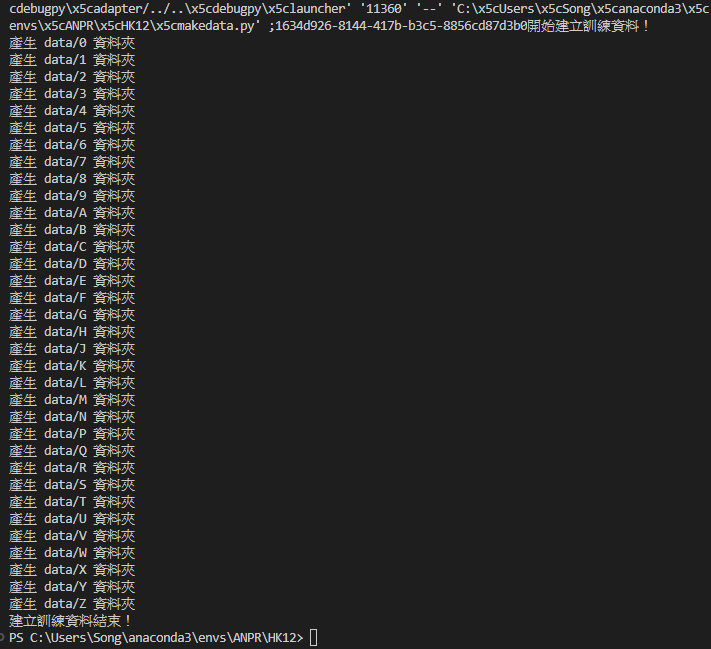
            cv2.imwrite('data/' + fontlist[n] + '/{:0>4d}.jpg'.format(index\*pic\_each+i+1), img)  #存檔

print('建立訓練資料結束！')

* *程式執行及說明:*

此程式首先定義了一個emptydir函數，用於清空並重新建立指定的資料夾。接著，在主程式中，為每個數字和字母建立一個存放訓練資料的目標資料夾data。然後，從來源資料夾platefont中讀取對應的字母和數字圖像，並通過在圖像中隨機加入雜點來生成多個變異圖像，最終將這些生成的圖像存入目標資料夾中。

* *程式輸出與結果描述:*



* *問題討論與程式修正:*

無

* *程式碼:*

train.py

import cv2

import os.path

import numpy as np

from imutils import paths

from sklearn.preprocessing import LabelBinarizer

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers.convolutional import Conv2D, MaxPooling2D

from keras.layers.core import Flatten, Dense

imagedir = "data"  #訓練資料

modelname = "carplate\_model.hdf5"  #模型名稱

data = []  #資料串列

labels = []  #標籤串列

#讀取資料

for image\_file in paths.list\_images(imagedir):

    image = cv2.imread(image\_file)

    image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  #轉為灰階

    label = image\_file.split(os.path.sep)[-2]  #擷取文字資料夾名稱

    data.append(image)  #加入圖形

    labels.append(label)  #加入標籤

data = np.array(data)  #轉換為numpy array

labels = np.array(labels)

#訓練資料佔85%

(X\_train, X\_test, Y\_train, Y\_test) = train\_test\_split(data, labels, test\_size=0.15, random\_state=0)

#標準化資料

X\_train\_normalize=X\_train.reshape(X\_train.shape[0],38,18,1).astype("float") / 255.0

X\_test\_normalize=X\_test.reshape(X\_test.shape[0],38,18,1).astype("float") / 255.0

#轉換標籤為one-hot

lb = LabelBinarizer().fit(Y\_train)

Y\_train\_OneHot = lb.transform(Y\_train)

Y\_test\_OneHot = lb.transform(Y\_test)

#建立模型

model = Sequential()

#神經網路

model.add(Conv2D(20, (5, 5), padding="same", input\_shape=(38, 18, 1), activation="relu"))

model.add(MaxPooling2D(pool\_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(50, (5, 5), padding="same", activation="relu"))

model.add(MaxPooling2D(pool\_size=(2, 2), strides=(2, 2)))

model.add(Flatten())

model.add(Dense(500, activation="relu"))

model.add(Dense(34, activation="softmax"))  #34類

model.compile(loss="categorical\_crossentropy", optimizer="adam", metrics=["accuracy"])

#開始訓練

model.fit(X\_train\_normalize, Y\_train\_OneHot, validation\_split=0.2, batch\_size=32, epochs=10, verbose=1)

model.save(modelname)  #儲存模型

#準確率

scores = model.evaluate(X\_train\_normalize , Y\_train\_OneHot)

print(scores[1])

scores2 = model.evaluate(X\_test\_normalize , Y\_test\_OneHot)

print(scores2[1])

* *程式執行及說明:*

1.資料讀取與處理：

* 從資料夾data中讀取所有圖像，轉換為灰階並提取標籤。
* 將圖像資料和標籤轉換為numpy array格式。

2.資料分割與標準化：

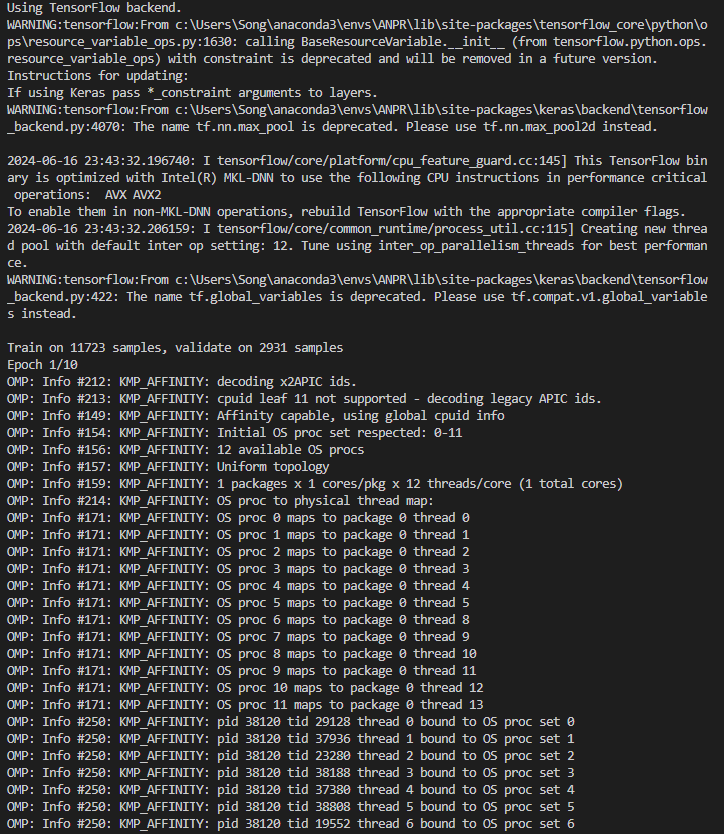
* 將資料分為訓練集（85%）和測試集（15%）。
* 將圖像資料標準化到0到1之間。
* 將標籤轉換為one-hot編碼。

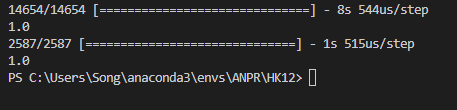
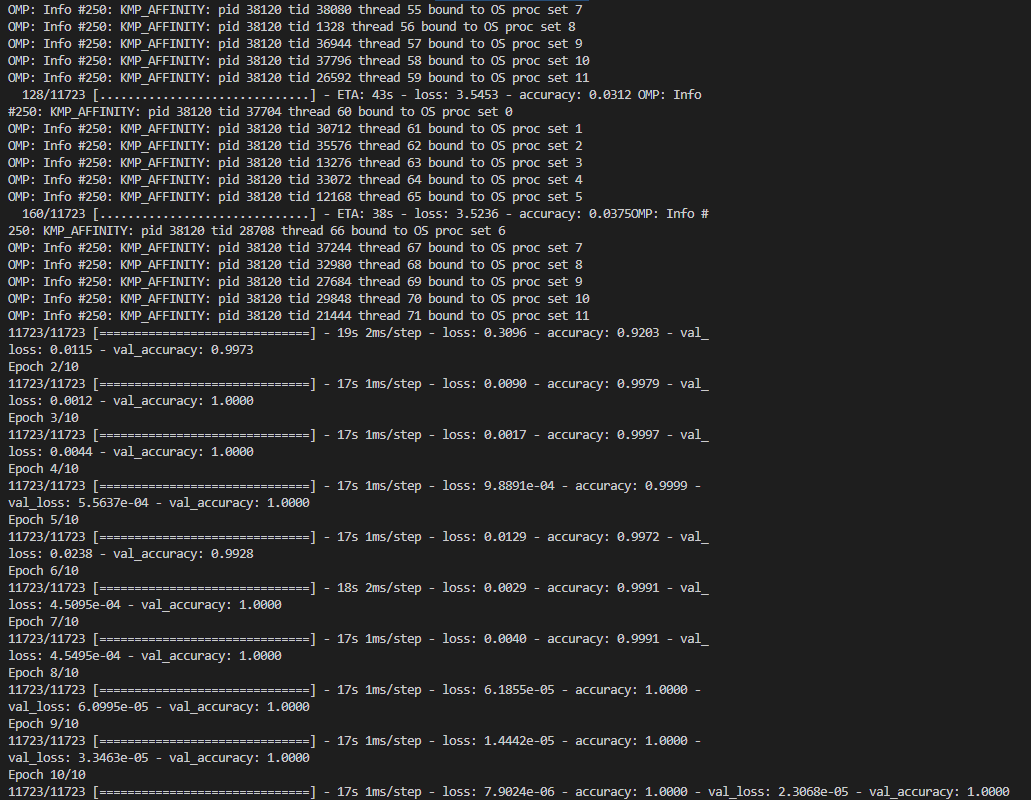
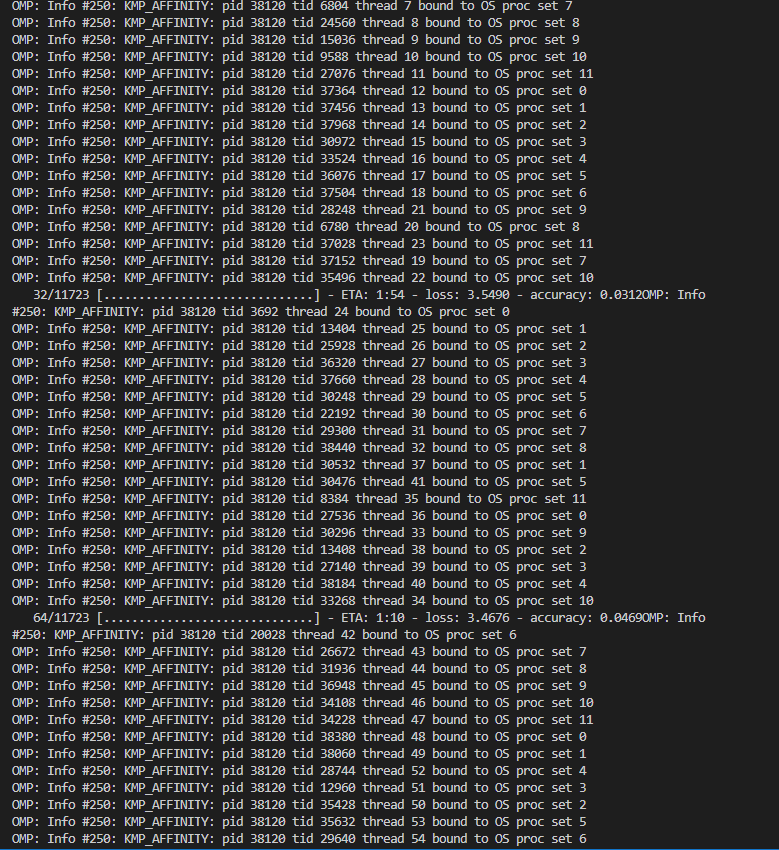
3.模型建立與訓練：

* 建立卷積神經網絡模型，包含兩層卷積層和池化層，最後是全連接層。
* 使用categorical\_crossentropy損失函數和adam優化器進行編譯。
* 訓練模型並儲存。

4.模型評估：

* 評估模型在訓練集和測試集上的準確率。
* *程式輸出與結果描述:*





* *問題討論與程式修正:*

無

* *程式碼:*

predict.py

from keras.models import load\_model

from PIL import Image

import numpy as np

import os

labels = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','J','K','L','M','N','P','Q','R','S','T','U','V','W','X','Y','Z']  #標籤值

datan = 0  #車牌文字數,即檔案數

for fname in os.listdir('cropMono'):

    if os.path.isfile(os.path.join('cropMono', fname)):

        datan += 1

tem\_data = []

for index in range(1, (datan+1)):  #讀取預測資料

    tem\_data.append((np.array(Image.open("cropMono/" + str(index) +".jpg")))/255.0)

real\_data = np.stack(tem\_data)  #(6,38,18)

real\_data1 = np.expand\_dims(real\_data, axis=3)  #轉換為(6,38,18,1)

model = load\_model("carplate\_model.hdf5")  #讀取模型

predictions = model.predict\_classes(real\_data1)  #預測資料

print('車牌號碼為：')

for i in range(len(predictions)):  #顯示結果

    print(labels[int(predictions[i])], end='')

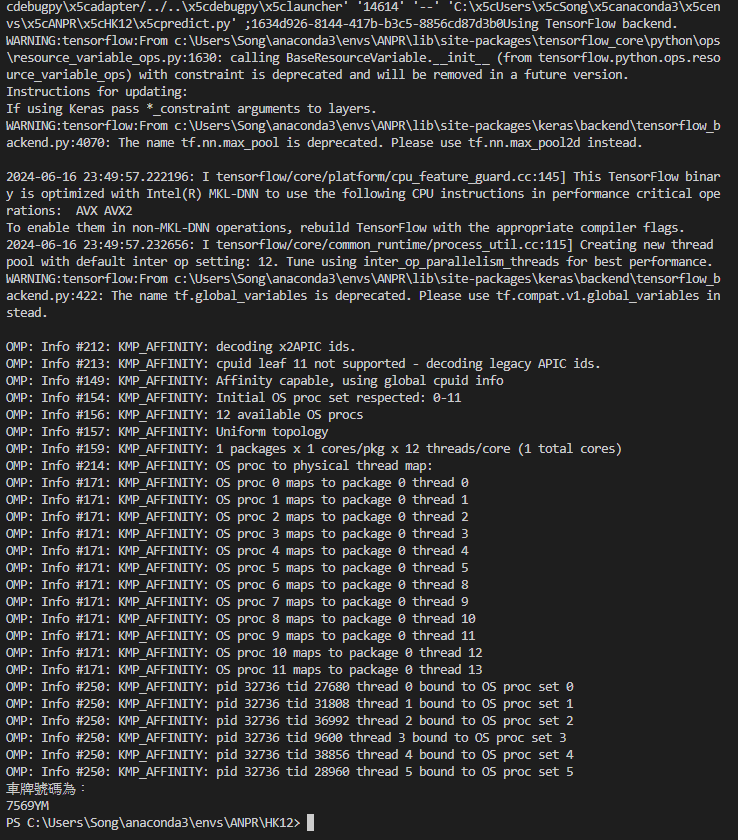
* *程式執行及說明:*

1.資料讀取：

* 從cropMono資料夾中讀取所有待預測的車牌字符圖像。
* 將圖像標準化並轉換為適合模型輸入的格式。

2.模型加載與預測：

* 加載訓練好的模型carplate\_model.hdf5。
* 使用模型對讀取的車牌字符進行預測。
* 根據預測結果顯示車牌號碼。
* *程式輸出與結果描述:*



* *問題討論與程式修正:*

第一次執行會遇到編碼問題，要去一個叫saving.py的程式把utf8那幾段刪掉。

* *程式碼:*

recogPlate\_all.py

def emptydir(dirname):  #清空資料夾

    if os.path.isdir(dirname):  #資料夾存在就刪除

        shutil.rmtree(dirname)

        sleep(2)  #需延遲,否則會出錯

    os.mkdir(dirname)  #建立資料夾

from keras.models import load\_model

from PIL import Image

import numpy as np

import cv2

import shutil, os

from time import sleep

import glob

labels = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','J','K','L','M','N','P','Q','R','S','T','U','V','W','X','Y','Z']  #標籤值

dirname = 'recogdata'

model = load\_model("carplate\_model.hdf5")  #讀取模型

myfiles = glob.glob('predictPlate/\*.jpg')

for imgname in myfiles:

    #擷取車牌

    emptydir(dirname)

    img = cv2.imread(imgname)

    detector = cv2.CascadeClassifier('haar\_carplate.xml')

    signs = detector.detectMultiScale(img, scaleFactor=1.1, minNeighbors=4, minSize=(20, 20))

    if len(signs) > 0 :

        for (x, y, w, h) in signs:

            image1 = Image.open(imgname)

            image2 = image1.crop((x, y, x+w, y+h))

            image3 = image2.resize((140, 40), Image.ANTIALIAS)

            image3.save('tem.jpg')

            image4 = cv2.imread('tem.jpg')

            gray = cv2.cvtColor(image4, cv2.COLOR\_RGB2GRAY)

            \_, img\_thre = cv2.threshold(gray, 100, 255, cv2.THRESH\_BINARY)

            cv2.imwrite('tem.jpg', img\_thre)

        #分割文字

        img\_tem = cv2.imread('tem.jpg')

        gray = cv2.cvtColor(img\_tem, cv2.COLOR\_BGR2GRAY)

        \_,thresh = cv2.threshold(gray, 127, 255, cv2.THRESH\_BINARY\_INV)  #轉為黑白

        contours1 = cv2.findContours(thresh.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  #尋找輪廓

        contours = contours1[1]  #取得輪廓

        letter\_image\_regions = []  #文字圖形串列

        for contour in contours:  #依序處理輪廓

            (x, y, w, h) = cv2.boundingRect(contour)  #單一輪廓資料

            letter\_image\_regions.append((x, y, w, h))  #輪廓資料加入串列

        letter\_image\_regions = sorted(letter\_image\_regions, key=lambda x: x[0])  #按X坐標排序

        #存檔

        i=1

        for letter\_bounding\_box in letter\_image\_regions:  #依序處理輪廓資料

            x, y, w, h = letter\_bounding\_box

            if w>=5 and h>32 and h<40:  #長度>6且高度在30-48才是文字

                letter\_image = gray[y:y+h, x:x+w]  #擷取圖形

                letter\_image = cv2.resize(letter\_image, (18, 38))

                cv2.imwrite(dirname + '/{}.jpg'.format(i), letter\_image)  #存檔

                i += 1

        #辨識車牌

        datan = 0  #車牌文字數,即檔案數

        for fname in os.listdir(dirname):

            if os.path.isfile(os.path.join(dirname, fname)):

                datan += 1

        tem\_data = []

        for index in range(1, (datan+1)):  #讀取預測資料

            tem\_data.append((np.array(Image.open("recogdata/" + str(index) +".jpg")))/255.0)

        real\_data = np.stack(tem\_data)

        real\_data1 = np.expand\_dims(real\_data, axis=3)  #轉換維度符合需求->(7,38,18,1)

        predictions = model.predict\_classes(real\_data1)  #預測資料

        result = []

        for i in range(len(predictions)):  #顯示結果

            result.append(labels[int(predictions[i])])

        print(imgname + ' --> ' + ''.join(result))

    else:

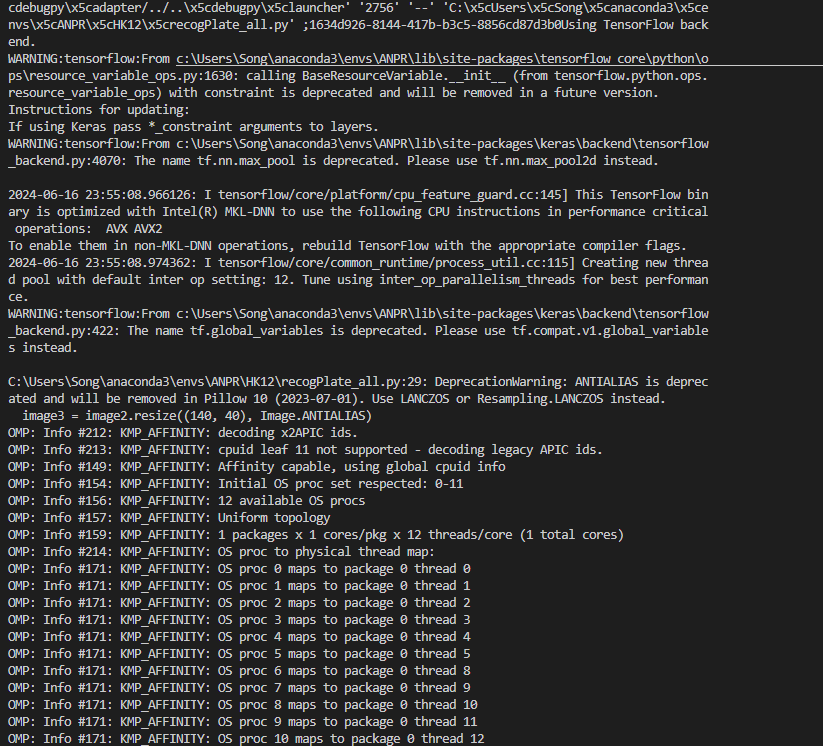
        print('無法擷取車牌！')

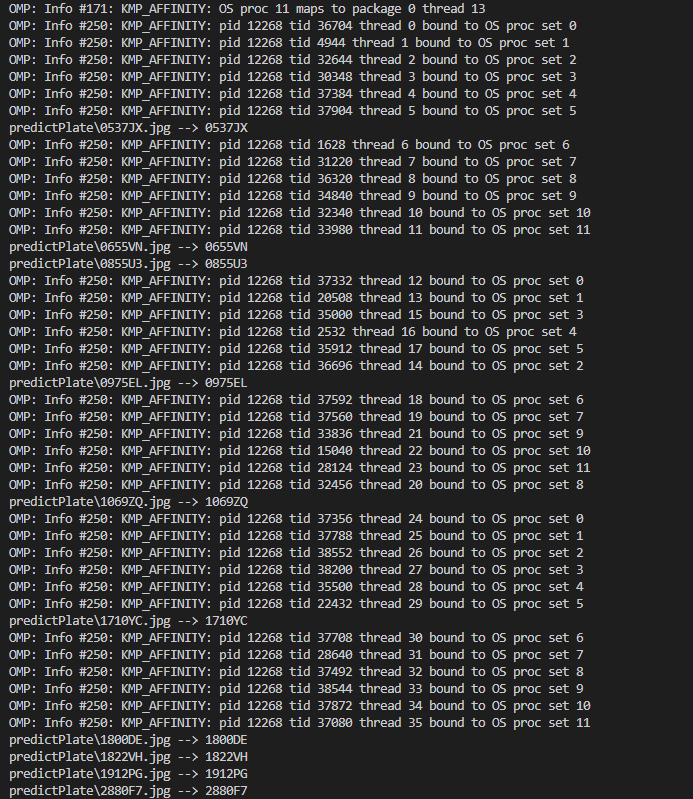
    os.remove('tem.jpg')

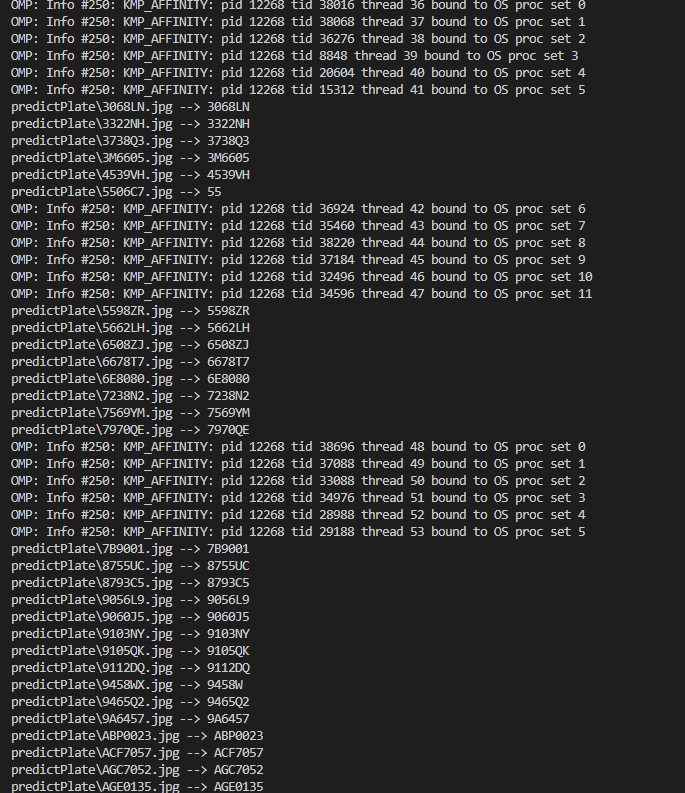
* *程式執行及說明:*

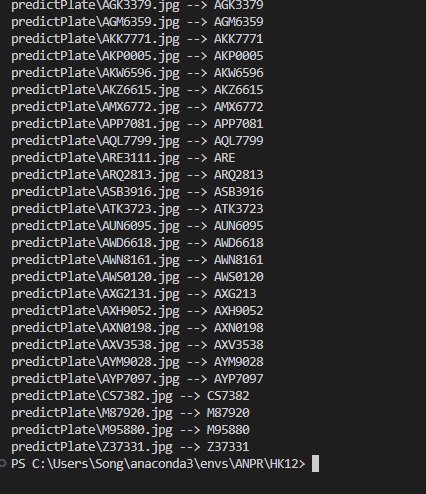
這段程式碼是一個用於辨識車牌的程式。它首先從一個資料夾中讀取待辨識的車牌圖片，然後使用 Haar 分類器擷取圖片中的車牌區域。接著對擷取到的車牌進行處理，包括裁剪、轉為灰度、二值化等操作，最後進行文字分割。分割後的文字圖像會被存儲到一個資料夾中，然後逐一讀取並使用事先訓練好的模型進行文字辨識。

* *程式輸出與結果描述:*









* *問題討論與程式修正:*

無