

驗證碼解析

資工系 410411804 張哲誠

資工系 410410632 李哲緯

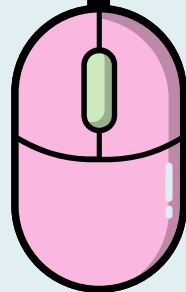
資工系 410411747 賴柏樟

企管系 409610242 徐欣瑜

企管系 409610366 洪若庭

21°

>>>>>





動機



.....

```
import cv2
import numpy as np
import tensorflow as tf
from tensorflow import keras

# def rectangle2square(img):
#     maxside = max(img.shape[0],img.shape[1])
#     white_img = np.zeros((maxside,maxside,1),np.uint8)
#     white_img.fill(255)
#     for x in range(img.shape[0]):
#         for y in range(img.shape[1]):
#             white_img[x,y] = img[x,y]
#     print(white_img.shape)
#     cv2.imshow("windows2",white_img)

# def use_model(img):
#     resize_img = rectangle2square(img)
#     resize_img = cv2.resize(resize_img, (28, 28))
#     # cv2.imshow("win", resize_img)
#     reshape_array = resize_img.flatten()
#     reshape_array = np.expand_dims(reshape_array, axis=0)
#     reshape_array = reshape_array.astype(np.float32) / 255
#     # print(reshape_array)
#     predictions = model.predict(reshape_array)
#     predicted_class_index = np.argmax(predictions)
#     print(predicted_class_index)
```



```
# model = keras.models.load_model("num_model.h5")
```



```
▼ def main():
```



直接以灰階圖讀取img

```
ori_img = cv2.imread("test.png", 0)
img = ori_img
```

將灰階圖的img_array變成one dimension

```
new_np_array = img.flatten()
```

得出img中最多的數值(背景顏色)

```
thresh_guess = np.argmax(np.bincount(new_np_array))
```



.....

print(thresh_guess) 印出重複最多的值
遍歷img_array 將重複最多的值設為白色 其餘黑色(白底黑字)

```
for x in range(img.shape[0]):  
    for y in range(img.shape[1]):  
        if img[x, y] == thresh_guess:  
            img[x, y] = 255  
        else:  
            img[x, y] = 0
```



用OTSU算法自動判別threshold

```
#thresh_value, img = cv2.threshold(img,1,255,cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)  
#print(thresh_value)
```

利用膨脹消除雜訊點及干擾線(白色背景增大)

```
kernel = np.ones([3, 3])  
img = cv2.dilate(img, kernel)
```

利用侵蝕將數字線條變寬

```
kernel = np.ones([4, 4])  
img = cv2.erode(img, kernel)
```

用Contours 把數字輪廓畫出

```
contours, _ = cv2.findContours(img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```



• • • • •

計算所有輪廓面積並將前四大的輸出(避免有沒清除的雜訊點被偵測)

```
areas = []
for c in contours:
    area = cv2.contourArea(c)
    areas.append(area)
areas = np.array(areas)
index = np.argsort(areas)[-5:]
top4_areas = []
for i in range(4):
    top4_areas.append(contours[index[i]])

# print(cv2.contourArea(top4_areas[-1]))
```



.....

利用boundingrect得到的x,y,w,h 傳進rectangle中並畫框

```
for each_num_contours in top4_areas:
    x, y, w, h = cv2.boundingRect(each_num_contours)
    cv2.rectangle(img, [x, y], [x + w, y + h], 2)

    # reconize_img = img[y : y + h, x : x + w]
    # reconize_img = rectangle2square(reconize_img)
    # use_model(reconize_img)

# cv2.imshow("windows", img)
# cv2.waitKey(0)

cv2.imwrite("result.png", img)

# return img
```

```
main()
```




GUI.PY



.....

```
import create_CAPTCHA
import main
```

圖片處理套件

```
from PIL import Image, ImageTk
```

GUI套件

```
import tkinter as tk
from tkinter import ttk
```



GUI.PY



• • • • •

輸出驗證碼圖片

```
▼ def create():  
    global img2
```

呼叫create_CAPTCHA.py生成驗證碼

```
create_CAPTCHA.createImg()  
img2 = Image.open('test.png')  
img2 = img2.resize(( 420, 200))  
img2 = ImageTk.PhotoImage( img2)
```

在output標籤輸出圖片

```
output.configure( image=img2)
```



GUI.PY



.....

輸出解析後的結果圖

```
▼ def result():  
    global img3
```

呼叫main.py生成結果圖

```
    main.main()  
    img3 = Image.open('result.png')  
    img3 = img3.resize(( 420, 200))  
    img3 = ImageTk.PhotoImage( img3)
```

在output標籤輸出圖片

```
    output.configure( image=img3)
```



GUI.PY

按下按鈕後執行並切換按鈕內容



.....

```
✓ def press_button():  
    if parse_text.get() == '產圖':  
        parse_text.set('解析')
```

驗證碼

```
        create()  
    else:  
        parse_text.set('產圖')
```

解析驗證碼結果

```
        result()
```

建立主視窗

```
root = tk.Tk()  
root.title('驗證碼解析')  
root.geometry('420x300')  
root.resizable( False, False)
```



GUI.PY

.....

風格設定、建立標籤顯示驗證圖/結果圖

```
style = ttk.Style()
style.theme_use('alt')
output = ttk.Label( root, font=('15'))
output.place( relx=0, rely=0.0, relwidth=1.0, relheight=0.7)

style.configure( 'TNotebook.Tab', font=('STHeiti', 20, 'bold'), padding=( 10, 5))
```

建立按鈕控制圖片輸出

```
parse_text = tk.StringVar()
parse_text.set('產圖')
parse = ttk.Button( root, textvariable=parse_text, padding=5)
parse.config( command=press_button)
parse.place( relx=0.5, rely=0.9, anchor='center')
```

分割線

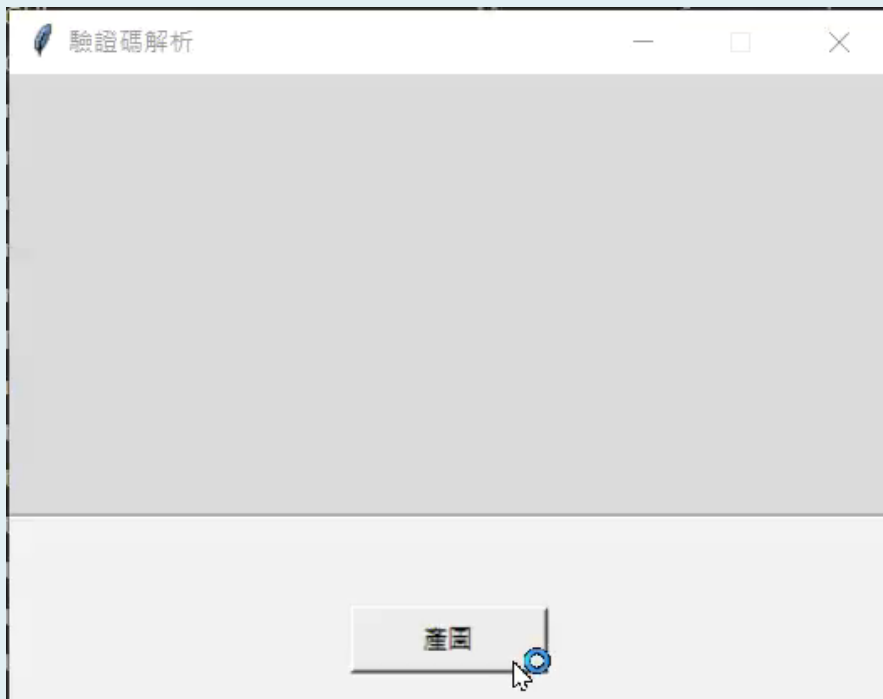
```
sep1 = ttk.Separator( root, orient='horizontal').place( rely=0.7, relwidth=1)
```

tkinter執行

```
root.mainloop()
```



實際操作影片



Thanks!

