



# 美肌處理

資工二C 吳信篁 409411351

2022/06/21

## 製作動機：

根據上課的內容，我覺得要做出有美肌功能的應用應該是非常實用的，所以選擇了美肌這個主題，並且加了一些上課所教的內容

## 功能介紹：

在程式中有設計 resize 來讓使用者調整到想要的圖片大小，接著進行高斯模糊處理以及雙邊濾波功能來使得圖片達到美肌的效果，然後使用者再從處理過後的圖片選擇一張自己喜歡的，接著進行儲存

## 工作原理：

在本程式中最主要用到的是 OPEN CV 函式中的 GaussianBlur、bilateralFilter

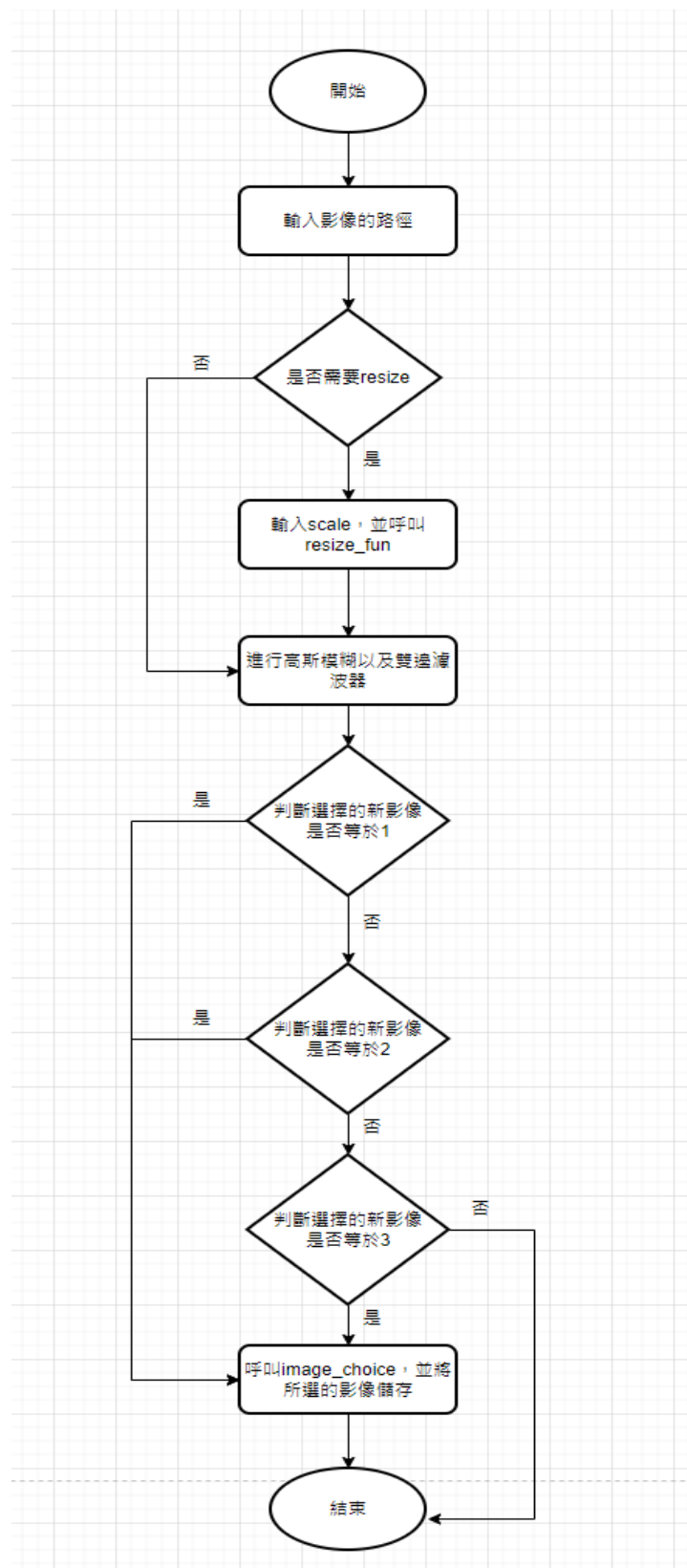
高斯模糊的原理：

類似於均值濾波（用周圍像素點的均值最為中心點的像素值），二維高斯函數濾波也是利用某點周圍的數值進行高斯模型處理，再將處理的數值作為該點的像素值，採用二維高斯函數如下， $G(x,y)=\frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}$ ，作為濾波器的係數

雙邊濾波器的原理：

採用加權平均的方法，用周邊像素亮度值的加權平均代表某個像素的強度，所用的加權平均基於高斯分佈，最重要的是，雙邊濾波的權重不僅考慮了像素的歐氏距離還考慮了像素範圍域中的輻射差異在計算中心像素的時候同時考慮這兩個權重

## 程式流程：



## 程式碼：

```
# import
import cv2
from matplotlib import pyplot as plt
import os

#定義 resize function
def resize_func(img, scale, interpolation):
    nr, nc = img.shape[:2]
    nr2 = int(nr*scale)
    nc2 = int(nc*scale)
    retImg = cv2.resize(img, (nc2, nr2), interpolation= interpolation)
    return retImg

#選擇的照片進行儲存
def image_choice(img, choice):
    print(f"\n\n\n\n\nYou Choice is {choice}, It shape is {img.shape}")
    #新增一個新的名稱給新的影像
    img_path = str(newImage_path)+'_newImg'+'.png'
    #將影像儲存
    cv2.imwrite(img_path, cv2.cvtColor(img, cv2.COLOR_BGR2RGB),
    [cv2.IMWRITE_JPEG_QUALITY, 90])
    print("Save Image succeed")

#輸入影像的相對路徑，並且取得影像圖檔的名稱
image_path = str(input("輸入影像路徑:"))
newImage_path = image_path[:len(image_path)-4]
img = cv2.imread(image_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
plt.show()

# 判斷是否需要進行resize
need_resize = eval(input("\n\n\n\n\n 如果要resize請選擇1 否則選2\n\nchoice:"))
#如果有需要輸入 scale 大小，並且呼叫resize_fun
```

```

if need_resize == 1:
    scale = eval(input("\n\n\n\n\n 請輸入 resize scale: "))
    img1 = resize_func(img, scale, cv2.INTER_CUBIC)
    print("Before image resize",img.shape)
    print("After image resize",img1.shape)
    img = img1
elif need_resize == 2:
    pass
#防呆處理
else:
    print(" 輸入有誤程式中止")
    os._exit(0)

#先用高斯模糊進行處理
guassian = cv2.GaussianBlur(img, (1,1), 0)

#再用雙邊濾波器進行美肌
newImg1 = cv2.bilateralFilter(guassian, 3, 50, 50)
newImg2 = cv2.bilateralFilter(guassian, 5, 50, 50)
newImg3 = cv2.bilateralFilter(guassian, 8, 50, 50)

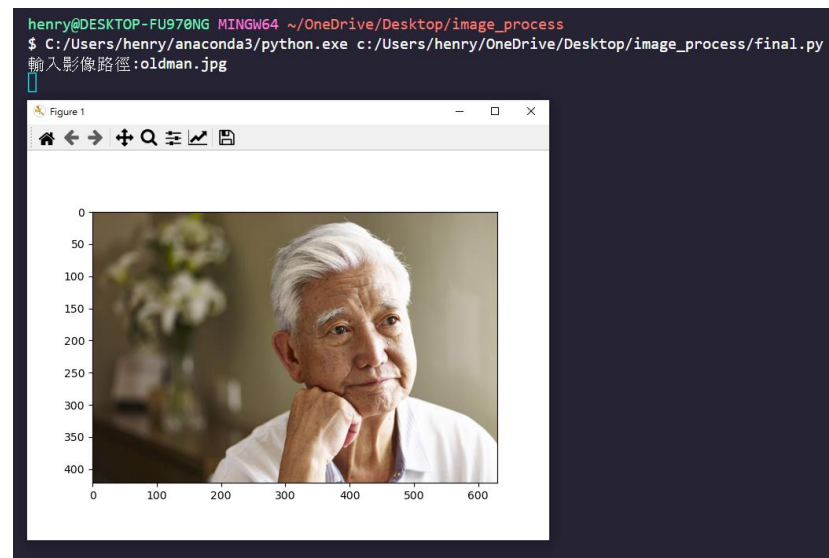
# show 出原圖以及進行過雙邊濾波器處理的影像
images = [img, newImg1, newImg2, newImg3]
titles = ['Original', 'Choice 1', 'Choice2', 'Choice3']
labels = ['', '3x3', '5x5', '8x8']
plt.figure(figsize=(20, 5))
for i in range(4):
    plt.subplot(1, 4, i+1),plt.imshow(images[i], cmap='gray')
    plt.title(titles[i], fontsize=15, color='r')
    plt.xlabel(labels[i], fontsize=15, color='b')
plt.tight_layout()
plt.show()

#選擇喜歡的影像，呼叫 image_choice 來進行儲存
img_choice = eval(input("\n\n\n\n\n 選擇你喜歡的:"))
if img_choice == 1:
    image_choice(newImg1, img_choice)

```

```
elif img_choice == 2:
    image_choice(newImg2, img_choice)
elif img_choice == 3:
    image_choice(newImg3, img_choice)
else:
    print("選擇不存在")
```

一開始輸入影像路徑

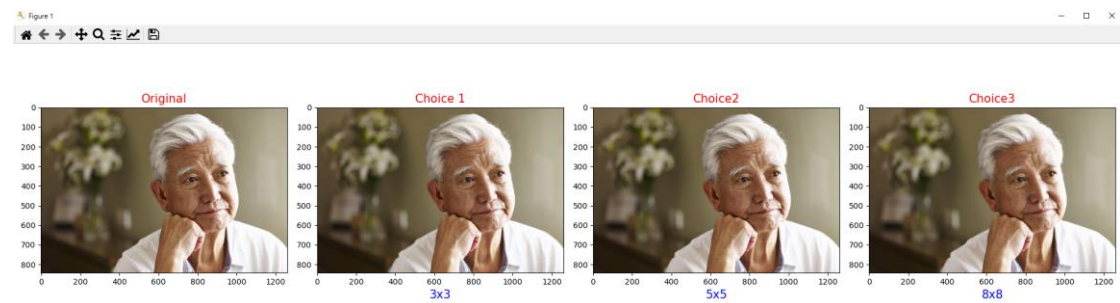


接著詢問是否需要 resize，這邊選擇 1(要)，並輸入 scale 大小

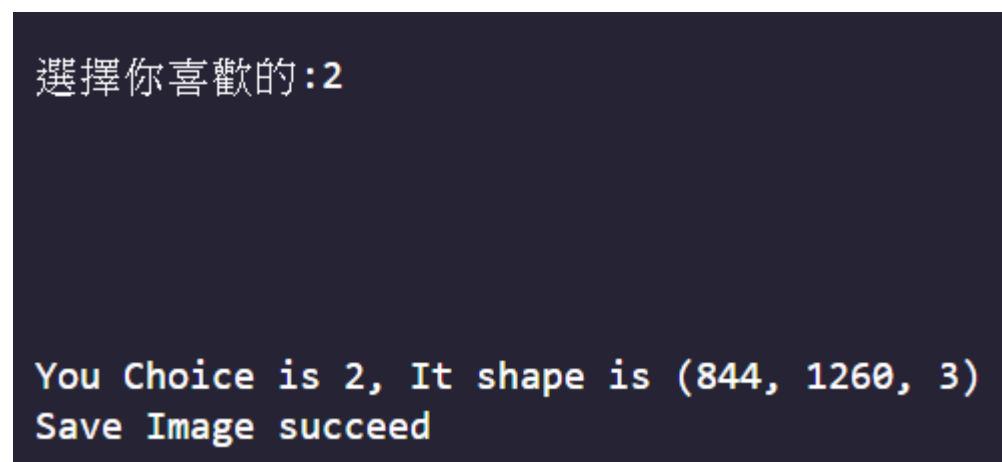
```
如果要resize請選擇1 否則選2
choice:1

請輸入resize scale: 2
Before image resize (422, 630, 3)
After image resize (844, 1260, 3)
█
```

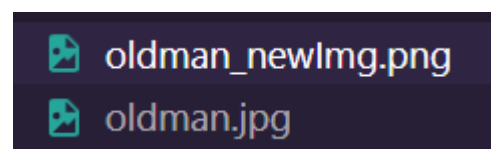
經過高斯模糊以及雙邊濾波器處理，出現處理過的圖片



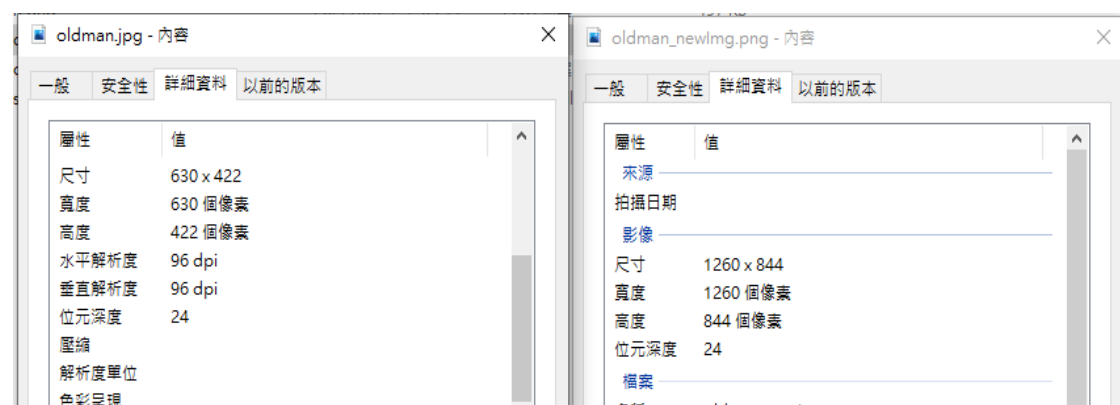
接著選擇喜歡的圖片，並進行儲存



在檔案出現一個新的 oldman\_newImg.png

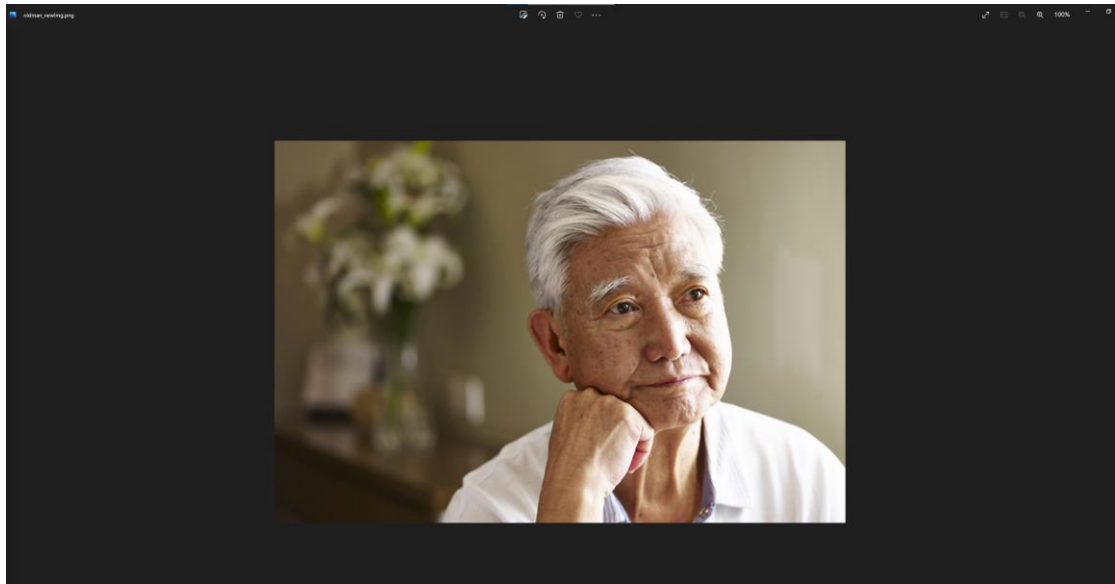


圖片確實有放大 2 倍





圖片也可以正常開啟



## 問題討論與心得：

在過程中有遇到一些難題，例如不到如何抓取圖片的檔名、resize 的參數調錯導致程式一直報錯、或者是防呆的處理該如何設計，雖然遇到一些奇奇怪怪的問題，但也都全部克服了，我覺得做這影像處理技術應用能讓我整合上課的內容，並從中挑選出我認為有趣的內容讓我自由發揮，是非常棒的一次經驗，但美中不足的是我認為本程式其實非常的單調，確實是可以再新增幾個有趣的功能，能讓程式變得更有趣且實用

## 成果影片：

<https://youtu.be/iD1bUyopZ5M>