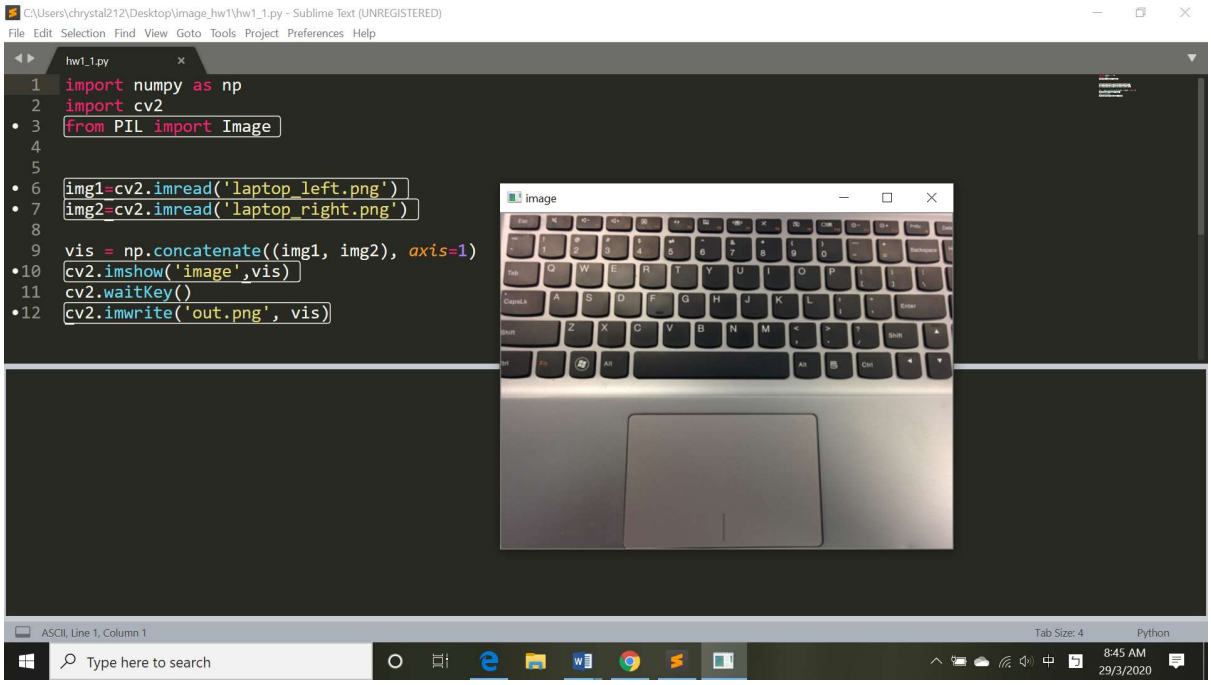
Image Processing – HW1

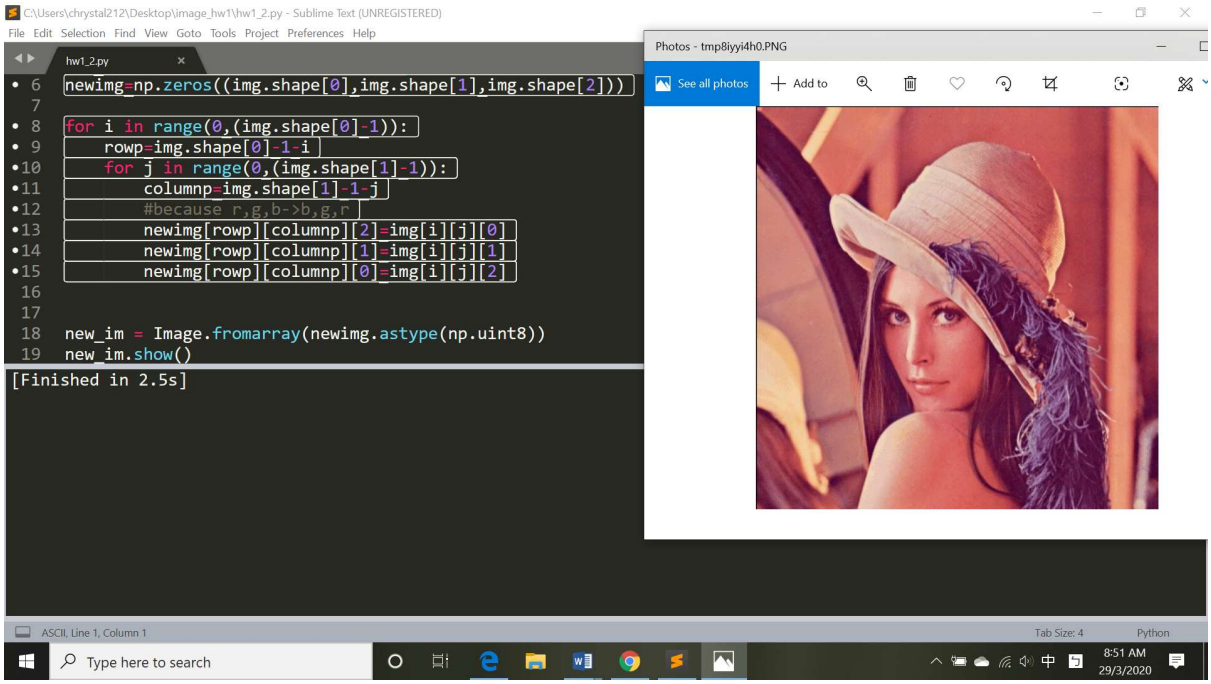
姓名:葉冠宏　學號:108753208　系級:資料所碩一

1.(25%) Create a program that combines two perfectly aligned pictures (laptop_left.png and laptop_right.png).



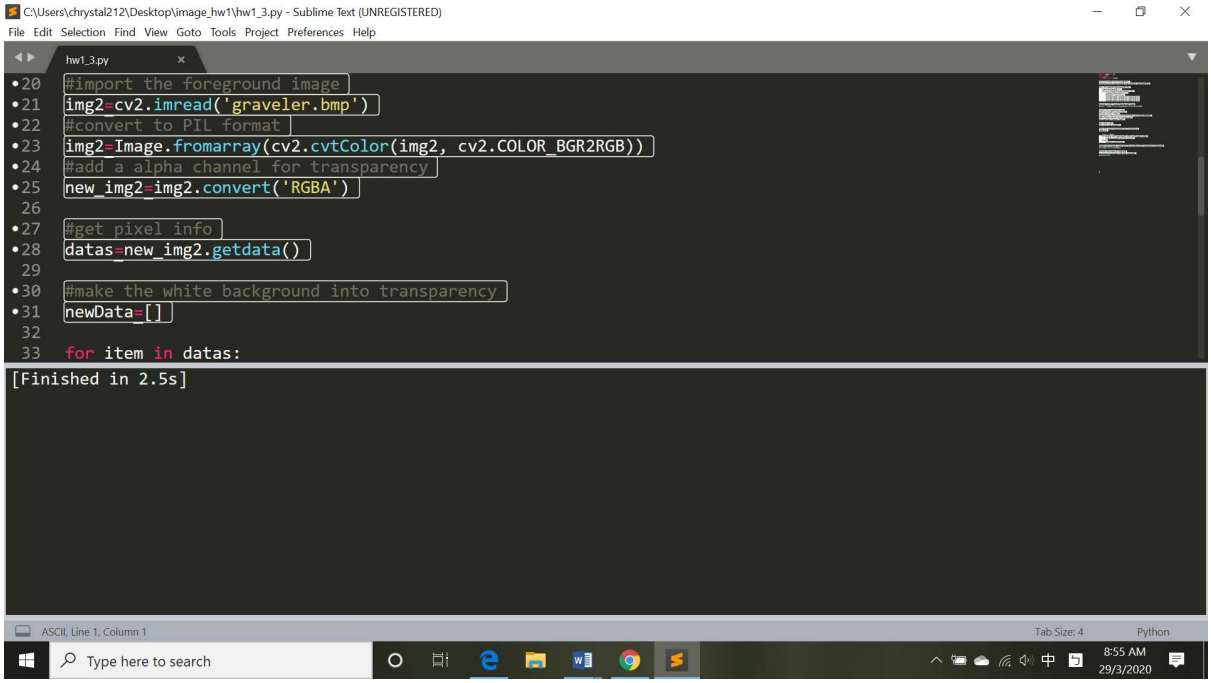
這邊我用 np.concatenate 把兩個 image 的矩陣去合併起來，所展現的成果如圖。

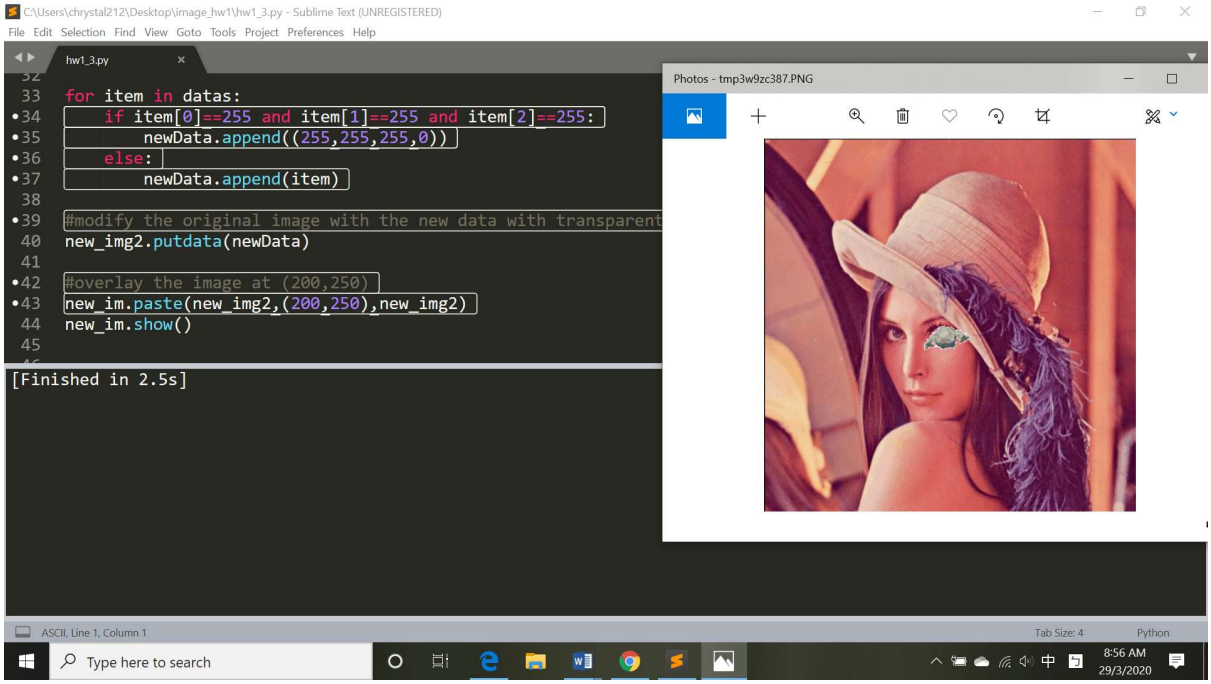
2.(25%) Implement a program to flip the image “lena_flipped.bmp”



這邊我創造一個矩陣去把原圖的矩陣的每一列的 i 去複製對應到全部列數-i，而每一行的 j 去複製對應到全部行數-j。這樣去做上下左右顛倒的效果。結果呈現如上。

3.(25%) Please overlay the image “graveler.bmp” without the white background onto the flipped lena image.





這邊我是針對“graveler.bmp”去增設 alpha 頻道去影響透明度的部分。我把白色背景的部分 alpha 值變為 0，即白色背景變成透明，剩下的部分透明度不變。最後用 paste 功能去 overlay 兩張圖片 。如圖眼睛的部分。

4.(25%, 10%, 15%)

(a) Please use a watermarking technique to embed “graveler.bmp” into the flipped lena image. You need to demonstrate how to embed and retrieve “graveler.bmp” from the image with the watermark.

(b) Please use the JPEG standard to encode the image with the watermark using different compression ratios (at least 3 different ratios), and decode it. Please check whether you can retrieve the watermark from the decoded image using the objective quality metric, PSNR.

(a)

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```python
def encode(im,im2):
    em=im2.shape
    im_c_=np.copy(im)

    for i in range(0,(em[0]-1)):
        for j in range(0,(em[1]-1)):
            c0=format(int(im_c_[i][j][0]),'08b')
            c1=format(int(im_c_[i][j][1]),'08b')
            c2=format(int(im_c_[i][j][2]),'08b')

            v0=format(int(im2[i][j][0]),'08b')
            v1=format(int(im2[i][j][1]),'08b')
            v2=format(int(im2[i][j][2]),'08b')
```

```
[Finished in 2.5s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:02 AM 29/3/2020

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```python
            r0=0
            r1=0
            r2=0

            for k in range(0,3):
                c0check=int(c0[k])
                c1check=int(c1[k])
                c2check=int(c2[k])
                power=7-k
                if(c0check==1):
                    r0=r0+pow(2,power)
                if(c1check==1):
                    r1=r1+pow(2,power)
                if(c2check==1):
```

```
[Finished in 2.5s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:02 AM 29/3/2020

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```python
                if(c2check==1):
                    r2=r2+pow(2,power)

            for m in range(0,3):
                v0check=int(v0[m])
                v1check=int(v1[m])
                v2check=int(v2[m])
                power2=3-m
                if(v0check==1):
                    r0=r0+pow(2,power2)
                if(v1check==1):
                    r1=r1+pow(2,power2)
                if(v2check==1):
                    r2=r2+pow(2,power2)
```

```
[Finished in 2.5s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:02 AM 29/3/2020

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

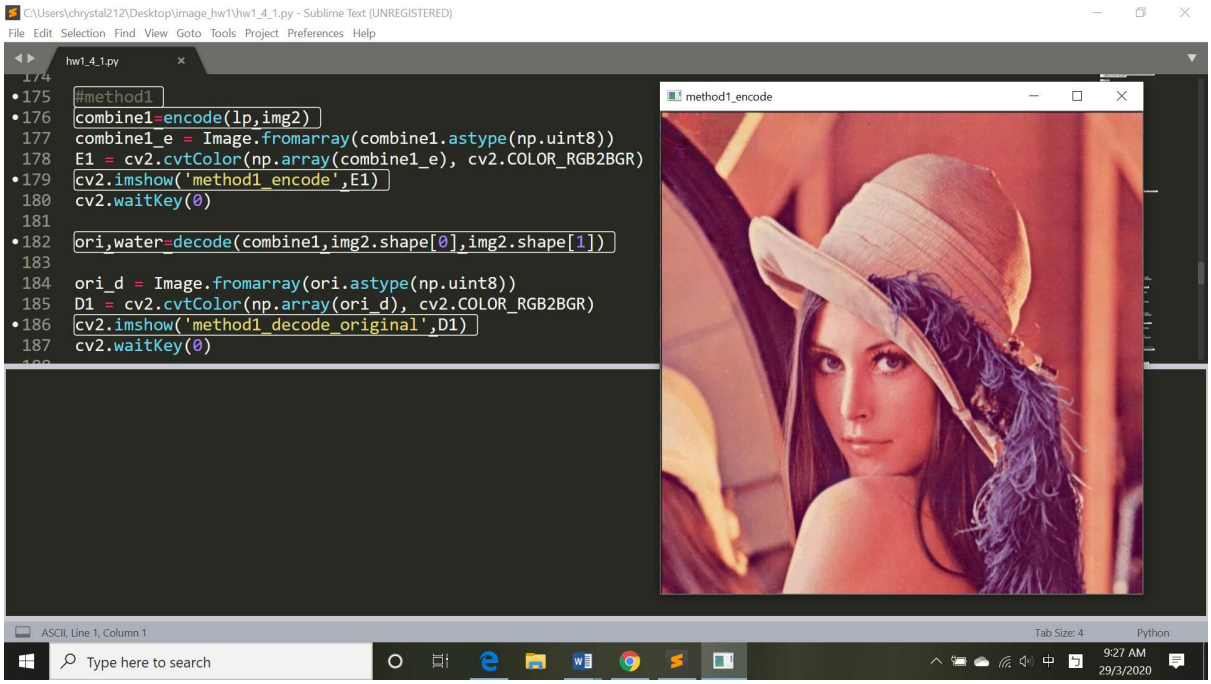
hw1_4_1.py

```python
                v2check=int(v2[m])
                power2=3-m
                if(v0check==1):
                    r0=r0+pow(2,power2)
                if(v1check==1):
                    r1=r1+pow(2,power2)
                if(v2check==1):
                    r2=r2+pow(2,power2)

            im_c_[i][j][0]=r0
            im_c_[i][j][1]=r1
            im_c_[i][j][2]=r2

    return im_c_
```

```
[Finished in 2.5s]
```

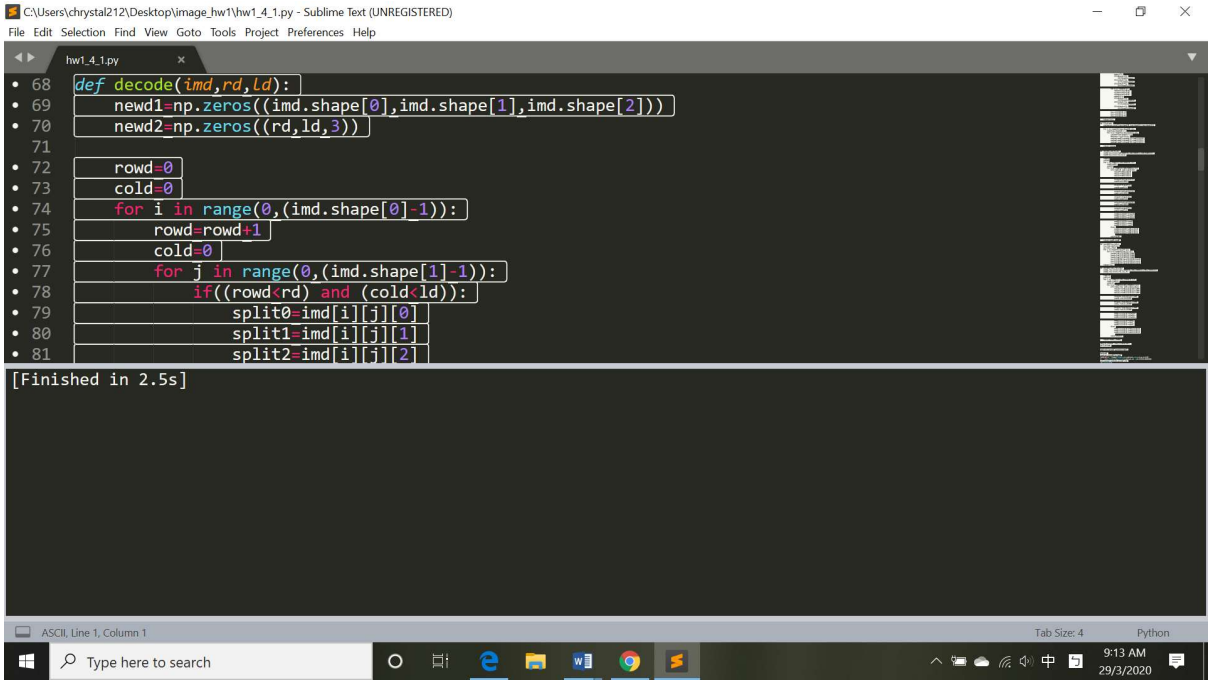
ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:02 AM 29/3/2020



上面的 encode function 在做的是，首先我們先去把傳入的 lena 圖用 np.copy 複製到 im_c_。然後，我們去取 lena image 要 overlay 的地方的 r,g,b 各先把數值轉換成 8-bit 的模式，然後我們再去取較高的四個位數來當作 lena image 的訊息。接著我們也去取 graveler.bmp 的 r,g,b 各先把數值轉換成 8-bit 的模式，然後我們去取其最高的四個位元的訊息，再去除以 16。我們把剛剛求得的兩個數值去相加，對 r,g,b 儲存到 r0,r1,r2，再去覆蓋 im_c_的 overlay 的地方的資訊。這樣就完成了把 watermark 的訊息隱藏到較低的四個位元，而不影響到原圖大致樣子的效果。這邊我們是把 overlay 的地方放在左上角(0,0) 的地方。

如上圖，是我們把 watermark 去 overlay 到 lena 後的成果。



C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```
                origin0=int(split0/16)
                sepa0=split0%16

                origin0=origin0*16
                sepa0=sepa0*16

                origin1=int(split1/16)
                sepa1=split1%16

                origin1=origin1*16
                sepa1=sepa1*16

                origin2=int(split2/16)
                sepa2=split2%16
```

```
[Finished in 2.5s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:13 AM 29/3/2020

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```
                sepa2=split2%16

                origin2=origin2*16
                sepa2=sepa2*16

                newd1[i][j][0]=origin0
                newd1[i][j][1]=origin1
                newd1[i][j][2]=origin2

                newd2[i][j][0]=sepa0
                newd2[i][j][1]=sepa1
                newd2[i][j][2]=sepa2
            else:
                newd1[i][j][0]=imd[i][j][0]
```

```
[Finished in 2.5s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:13 AM 29/3/2020

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```
                newd1[i][j][1]=origin1
                newd1[i][j][2]=origin2

                newd2[i][j][0]=sepa0
                newd2[i][j][1]=sepa1
                newd2[i][j][2]=sepa2
            else:
                newd1[i][j][0]=imd[i][j][0]
                newd1[i][j][1]=imd[i][j][1]
                newd1[i][j][2]=imd[i][j][2]

            cold=cold+1

    return newd1,newd2
```

```
[Finished in 2.5s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:14 AM 29/3/2020

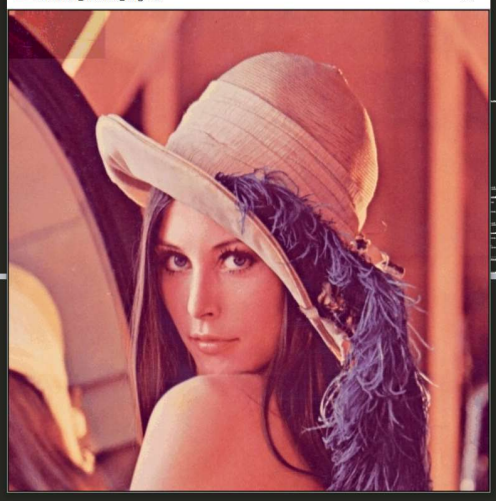
C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```
#method1
combine1=encode(lp,img2)
combine1_e = Image.fromarray(combine1.astype(np.uint8))
E1 = cv2.cvtColor(np.array(combine1_e), cv2.COLOR_RGB2BGR)
cv2.imshow('method1_encode',E1)
cv2.waitKey(0)

ori,water=decode(combine1,img2.shape[0],img2.shape[1])

ori_d = Image.fromarray(ori.astype(np.uint8))
D1 = cv2.cvtColor(np.array(ori_d), cv2.COLOR_RGB2BGR)
cv2.imshow('method1_decode_original',D1)
cv2.waitKey(0)
```

method1_decode_original



ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:28 AM 29/3/2020

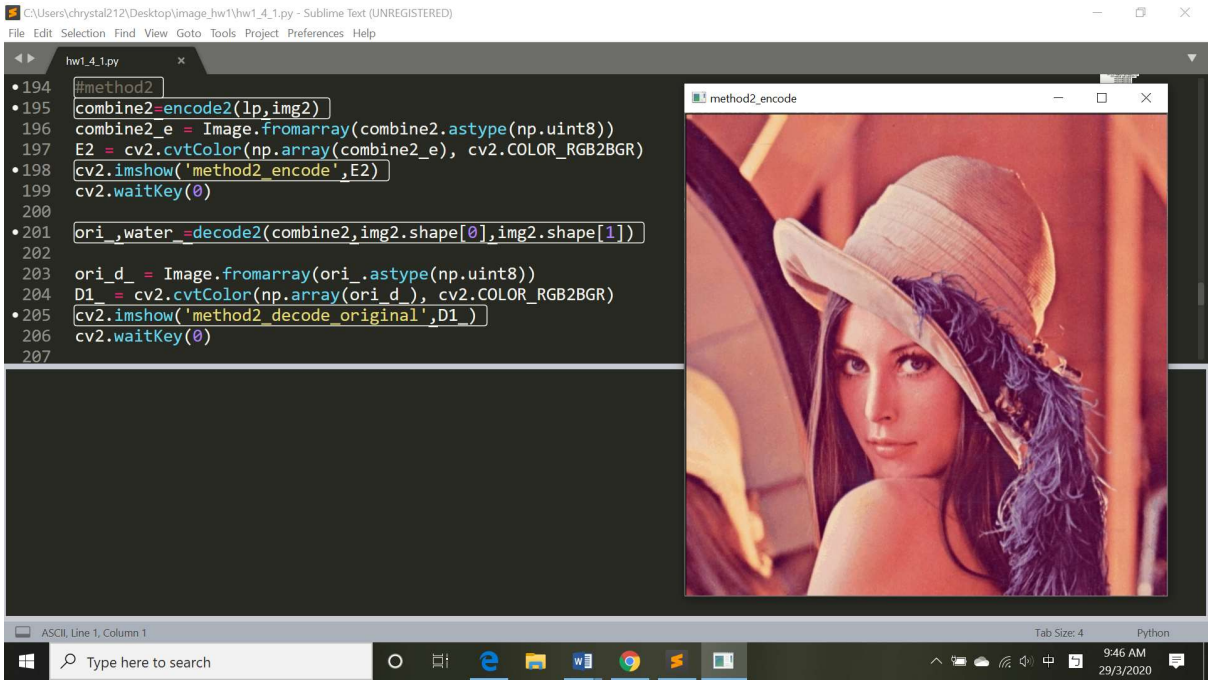
```python
#method1
combine1=encode(lp,img2)
combine1_e = Image.fromarray(combine1.astype(np.uint8))
E1 = cv2.cvtColor(np.array(combine1_e), cv2.COLOR_RGB2BGR)
cv2.imshow('method1_encode',E1)
cv2.waitKey(0)

ori,water=decode(combine1,img2.shape[0],img2.shape[1])

ori_d = Image.fromarray(ori.astype(np.uint8))
D1 = cv2.cvtColor(np.array(ori_d), cv2.COLOR_RGB2BGR)
cv2.imshow('method1_decode_original',D1)
cv2.waitKey(0)
```

在上面的 decode function 中，我們去把疊合的圖 imd，及 watermark 的 row 跟 column 參數傳入。接著我們去創造兩個 np.array，分別是 newd1，newd2，分別是 retrieve 後的 lena 及 watermark 的矩陣。我們的做法是，針對 overlay 的地方的 r,g,b 的數值 split0,split1,split2 分別去除以 16，取整數後分別得到 origin0,origin1,origin2，然後再去分別乘以 16 來當作 retrieve 後 lena 數值。而如果我們對 overlay 的地方的 r,g,b 的數值 split0,split1,split2 分別去 mod 16，取餘數得到 sepa0,sepa1,sepa2，再去分別乘以 16，就可以去還原當初 encode 前 watermark 的數值轉成二進位的最高 4 位元。我們用這樣的方法來去 retrieve “graveler.bmp” from the image with the watermark。

上面最後兩張圖分別是在 embed “graveler.bmp” into the flipped lena image 後，我們去 retrieve “graveler.bmp” from the image with the watermark 取出後 lena image 和 watermark 的樣子。

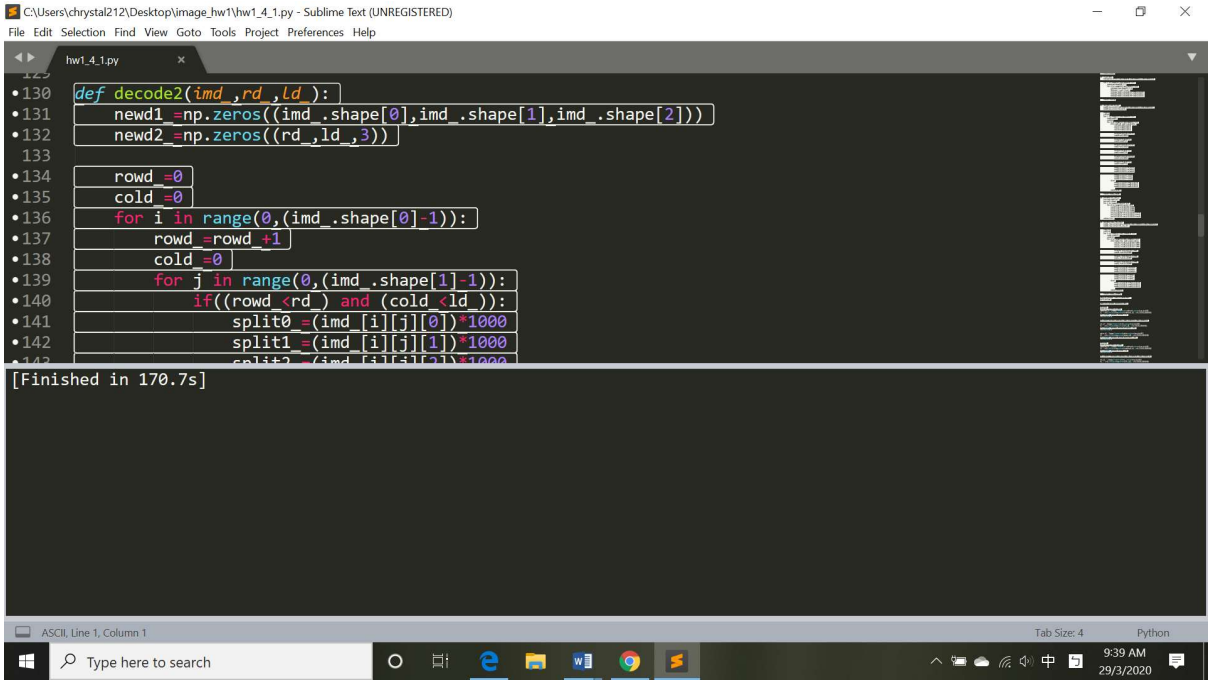
```python
def encode2(im_,im2_):
    im_c=np.copy(im_)
    em_=im2_.shape
    for i in range(0,(em_[0]-1)):
        for j in range(0,(em_[1]-1)):
            inse0=(im2_[i][j][0])/1000
            inse1=(im2_[i][j][1])/1000
            inse2=(im2_[i][j][2])/1000
            im_c[i][j][0]=im_c[i][j][0]+inse0
            im_c[i][j][1]=im_c[i][j][1]+inse1
            im_c[i][j][2]=im_c[i][j][2]+inse2
    return im_c
```

```
[Finished in 170.7s]
```



這邊，我也試著用其他方法去 encode 看看，當作 method2。作法是去把 watermark 的數值除以 1000，成為小數點後去加到 lena image 要 overlay 的地方的小數部分。這樣就可以更完整的保留 lena image 的資訊及 watermark 的資訊。

呈現 overlay 的效果如上。



C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```python
        for j in range(0,(imd_.shape[1]-1)):
            if((rowd_<rd_) and (cold_<ld_)):
                split0_=(imd_[i][j][0])*1000
                split1_=(imd_[i][j][1])*1000
                split2_=(imd_[i][j][2])*1000

                origin0_=int(split0_/1000)
                sepa0_=split0_%1000

                origin1_=int(split1_/1000)
                sepa1_=split1_%1000

                origin2_=int(split2_/1000)
                sepa2_=split2_%1000
```

```
[Finished in 170.7s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:39 AM 29/3/2020

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

```
                newd1_[i][j][0]=origin0_
                newd1_[i][j][1]=origin1_
                newd1_[i][j][2]=origin2_

                newd2_[i][j][0]=sepa0_
                newd2_[i][j][1]=sepa1_
                newd2_[i][j][2]=sepa2_
            else:
                newd1_[i][j][0]=imd_[i][j][0]
                newd1_[i][j][1]=imd_[i][j][1]
                newd1_[i][j][2]=imd_[i][j][2]

            cold_=cold_+1
```

```
[Finished in 170.7s]
```

ASCII, Line 1, Column 1 Tab Size: 4 Python

Type here to search 9:39 AM 29/3/2020

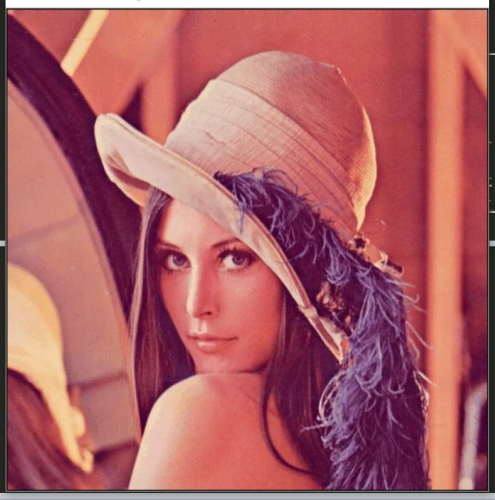
C:\Users\chrystal212\Desktop\image_hw1\hw1_4_1.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

hw1_4_1.py

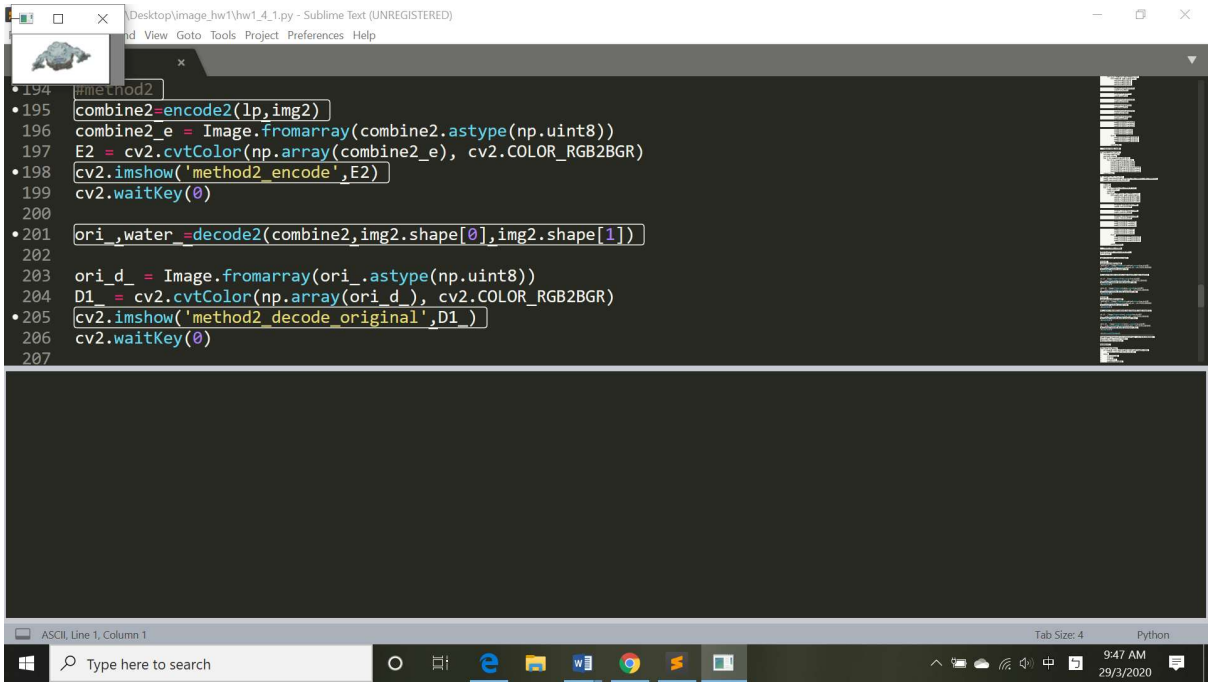
```
#method2
combine2=encode2(lp,img2)
combine2_e = Image.fromarray(combine2.astype(np.uint8))
E2 = cv2.cvtColor(np.array(combine2_e), cv2.COLOR_RGB2BGR)
cv2.imshow('method2_encode',E2)
cv2.waitKey(0)

ori_,water_=decode2(combine2,img2.shape[0],img2.shape[1])

ori_d_ = Image.fromarray(ori_.astype(np.uint8))
D1_ = cv2.cvtColor(np.array(ori_d_), cv2.COLOR_RGB2BGR)
cv2.imshow('method2_decode_original',D1_)
cv2.waitKey(0)
```

method2_decode_original



ASCII, Line 1, Column 1 Tab Size: 4 Python

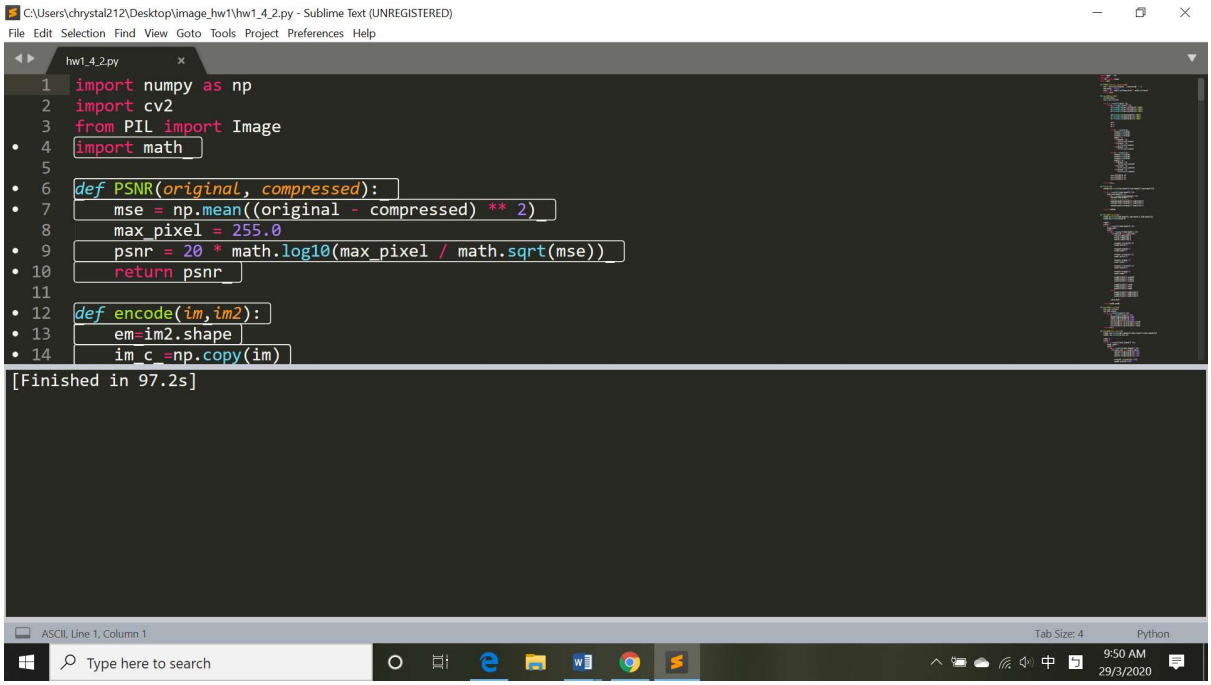
Type here to search 9:47 AM 29/3/2020



所以法二 decode 的部分，我們就只要對 overlay 的地方的 r,g,b 數值先分別乘以 1000，得到 split0_,split1_,split2_，然後再去除以 1000，取其整數部分來當作 retrieve 後 lena 的 image 資訊。然後對 split0_,split1_,split2 去 mod 1000 取餘數來當作 watermark 的資訊。以此演算法來 retrieve “graveler.bmp” from the image with the watermark.

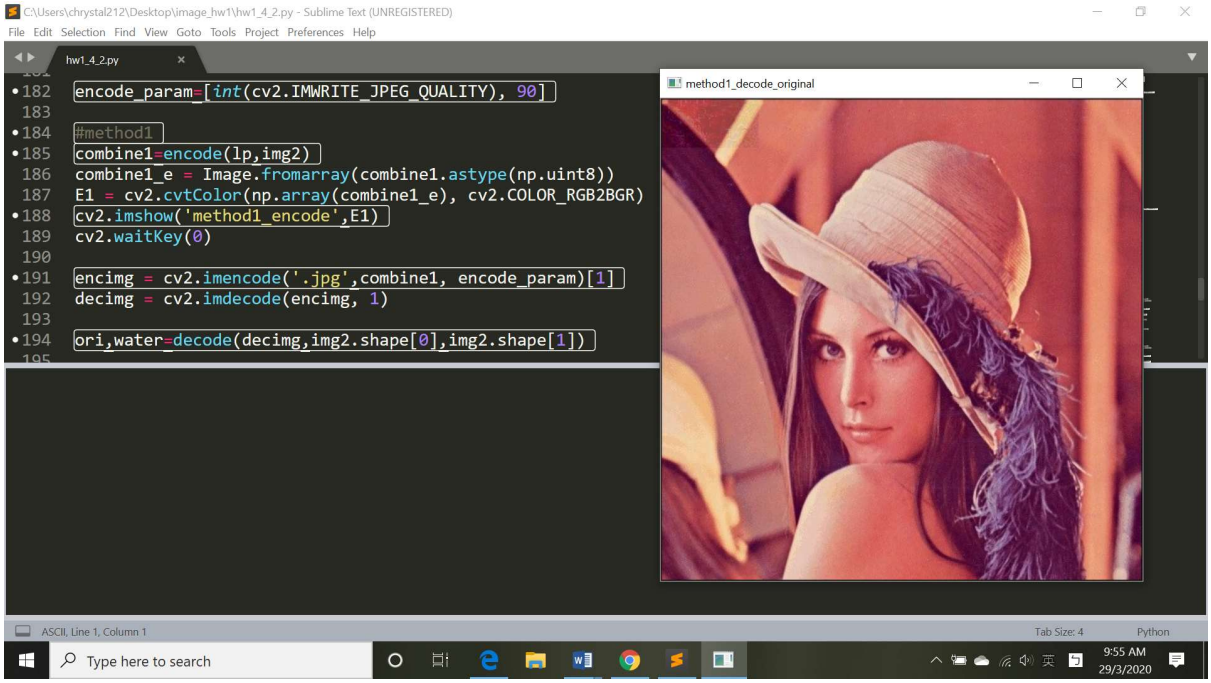
呈現的效果如最後兩張圖。法二完整保留了 watermark overlay 前的資訊。

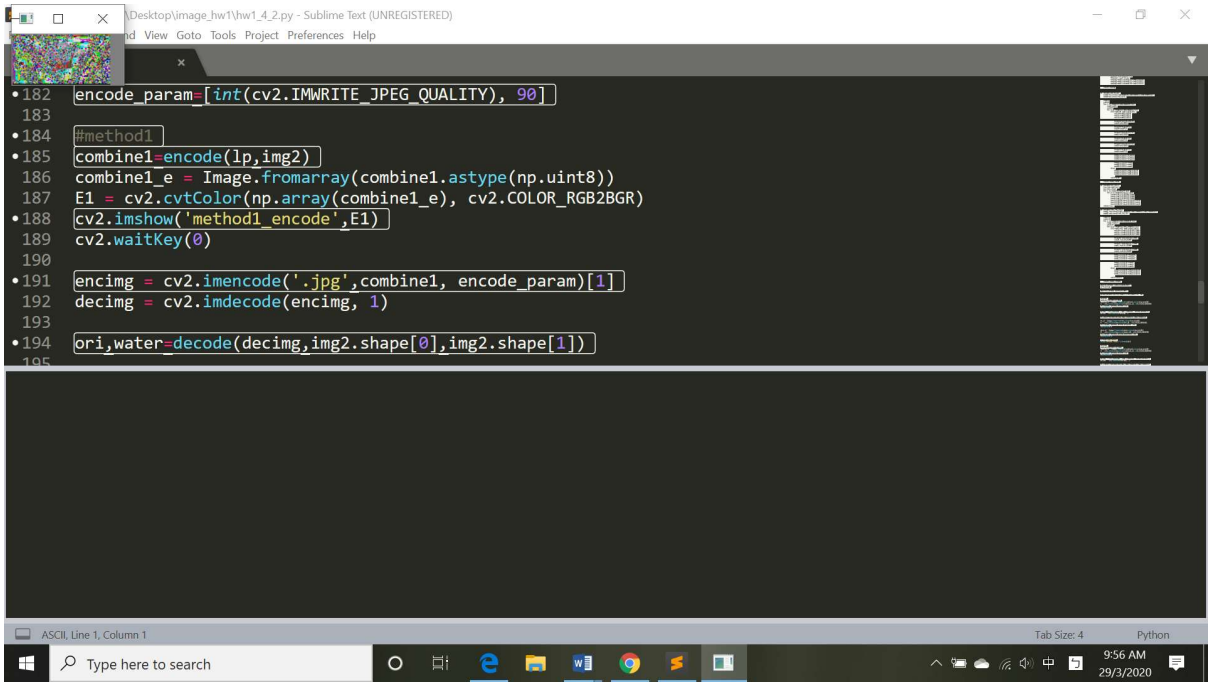
(b) Please use the JPEG standard to encode the image with the watermark using different compression ratios (at least 3 different ratios), and decode it. Please check whether you can retrieve the watermark from the decoded image using the objective quality metric, PSNR.

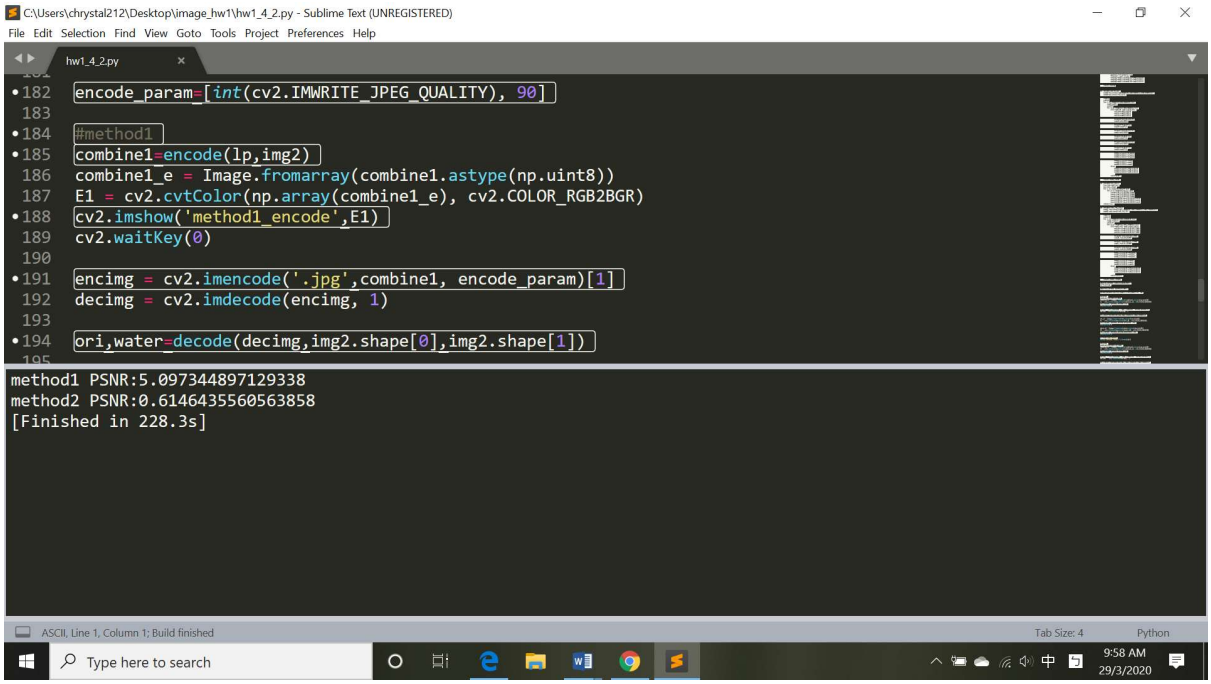


這邊是我 PSNR 的 function。

公式來源: https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/

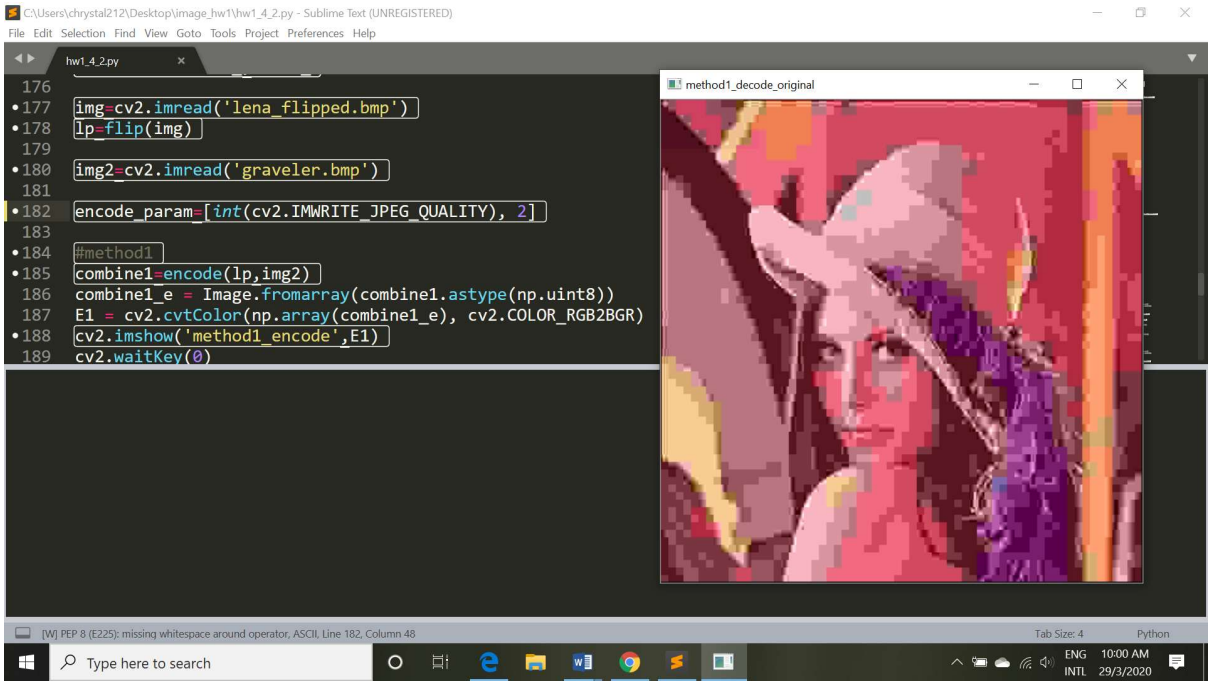


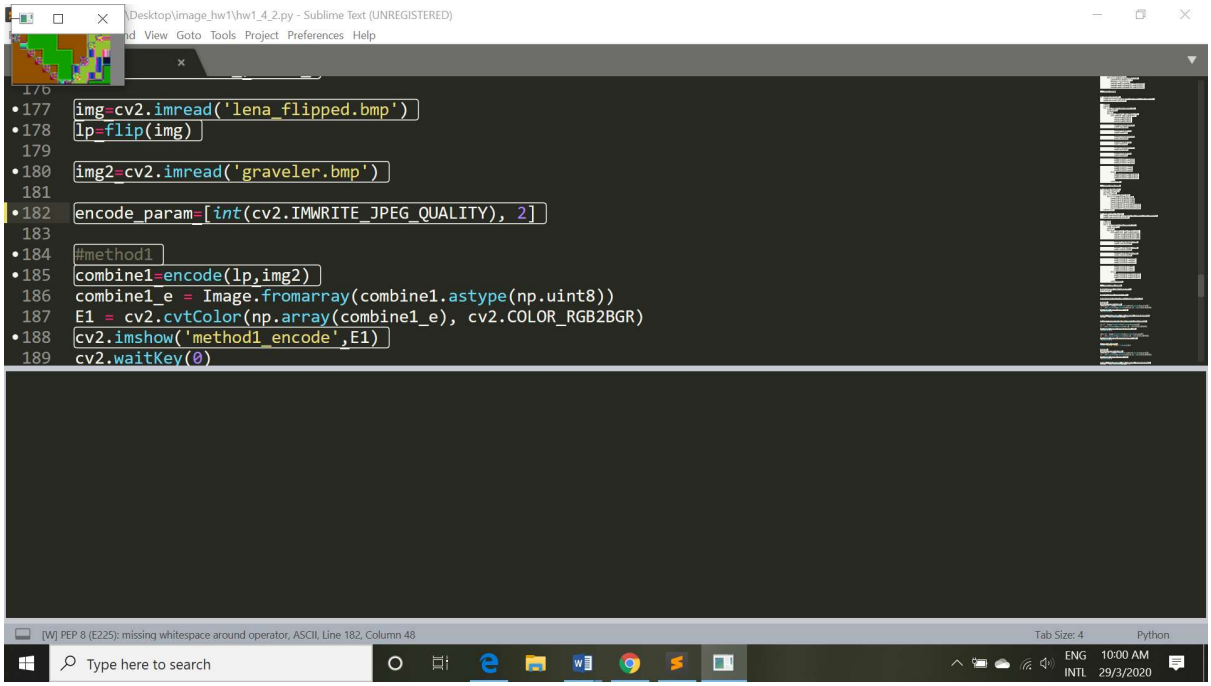


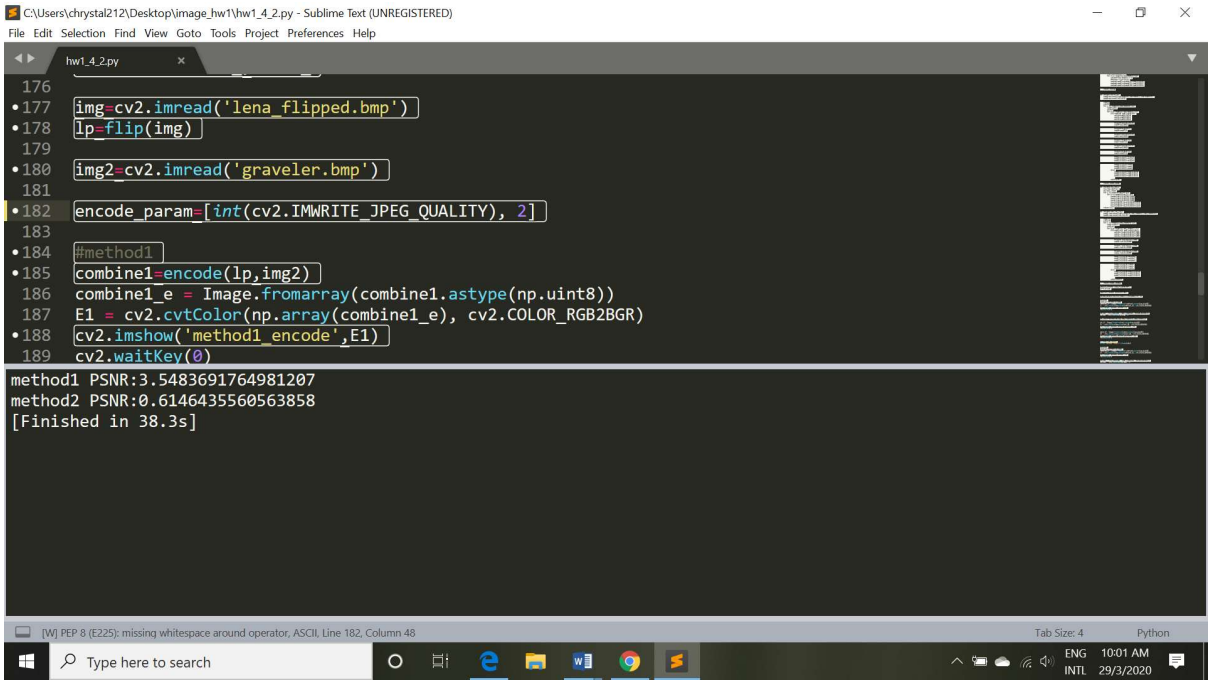


這邊第一個 compression ratio 是 90。我用 cv2.IMWRITE_JPEG_QUALITY 及 imencode,imdecode 的 function 來完成。

可以看到在壓縮完又還原後，我們去 retrieve watermark 及 lena image。我們看到 lena image 似乎不太有變化，而 watermark 的圖多了很多雜訊。而你所得到的 PSNR 值是 5.09 左右。

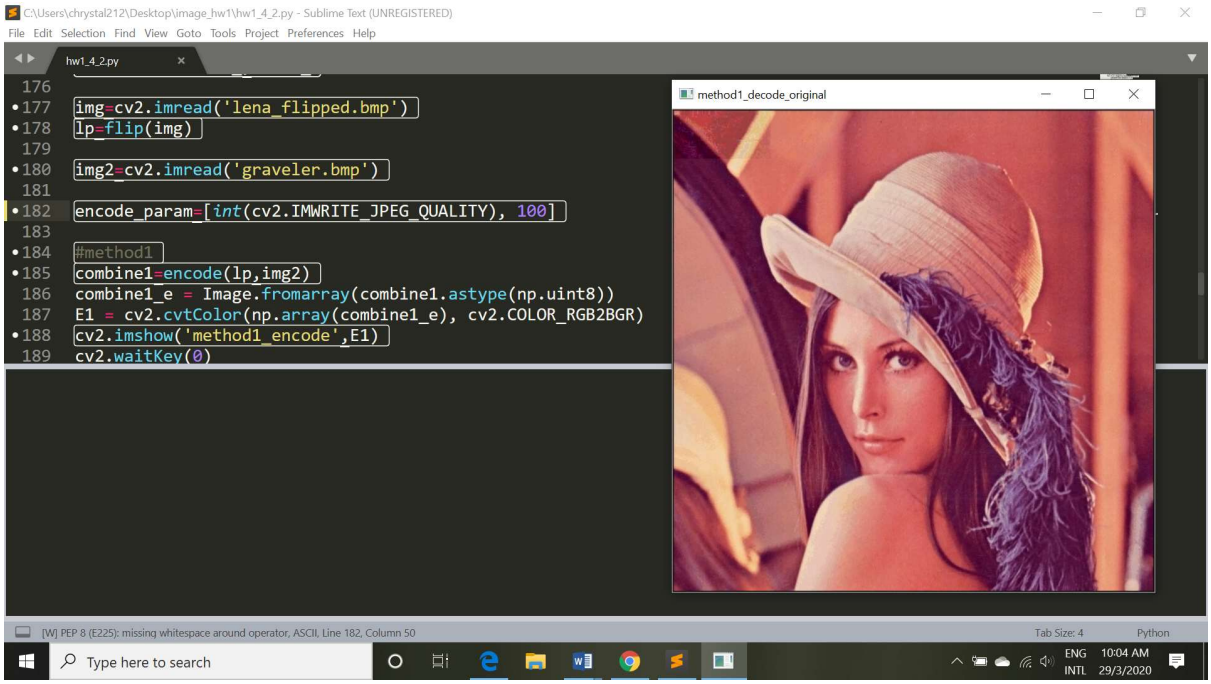


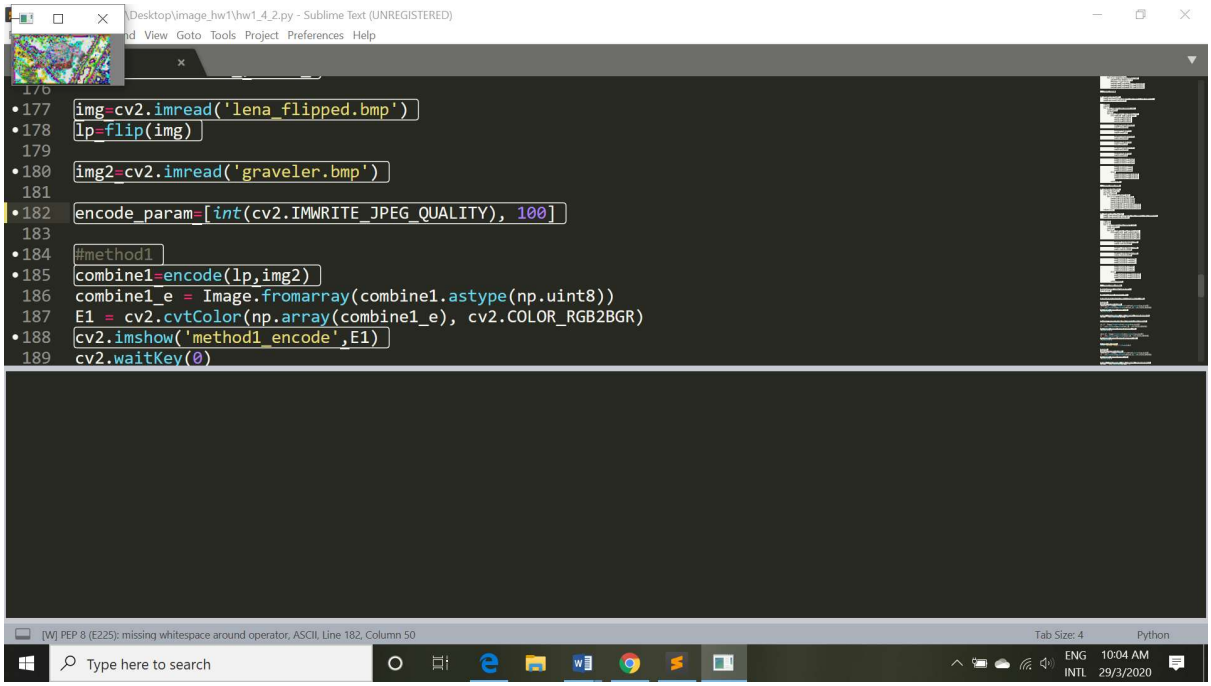


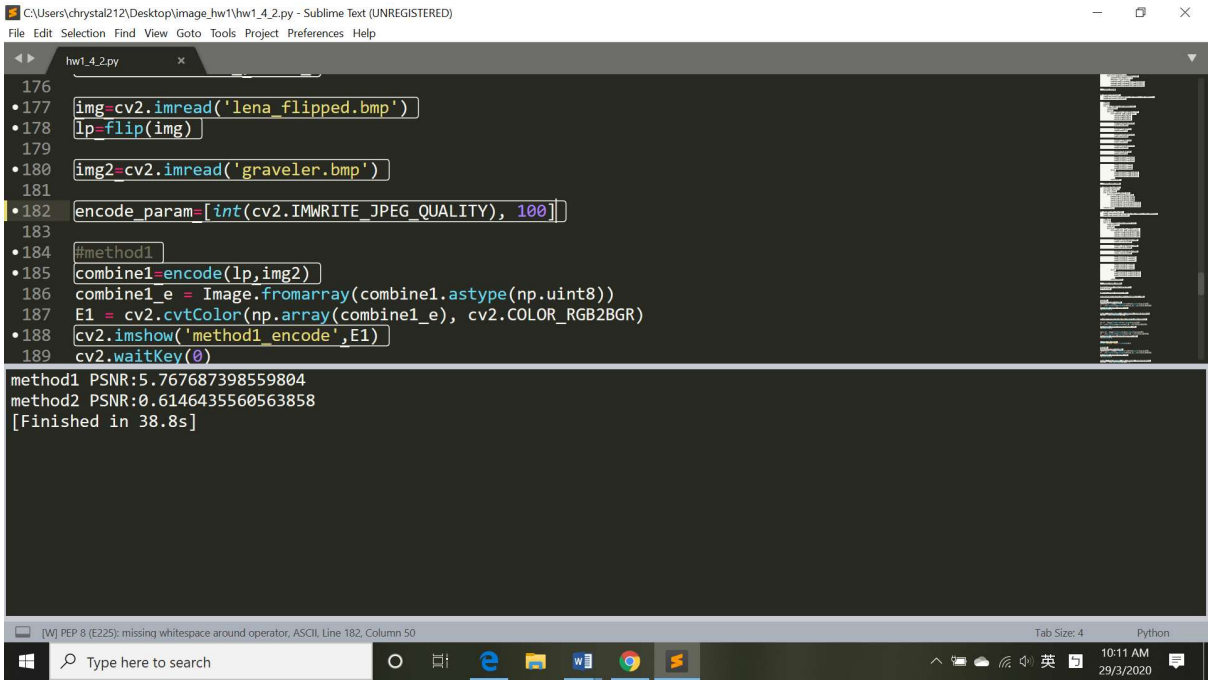


接著，我們用 compression ratio 是 2 來做。

可以看到在壓縮完又還原後，我們去 retrieve watermark 及 lena image。我們看到 lena image 的解析度也低很多，而 watermark 的圖則是變成看不出原本的樣貌。而你所得到的 PSNR 值是 3.548 左右，比 compression rate 為 90 時低很多。

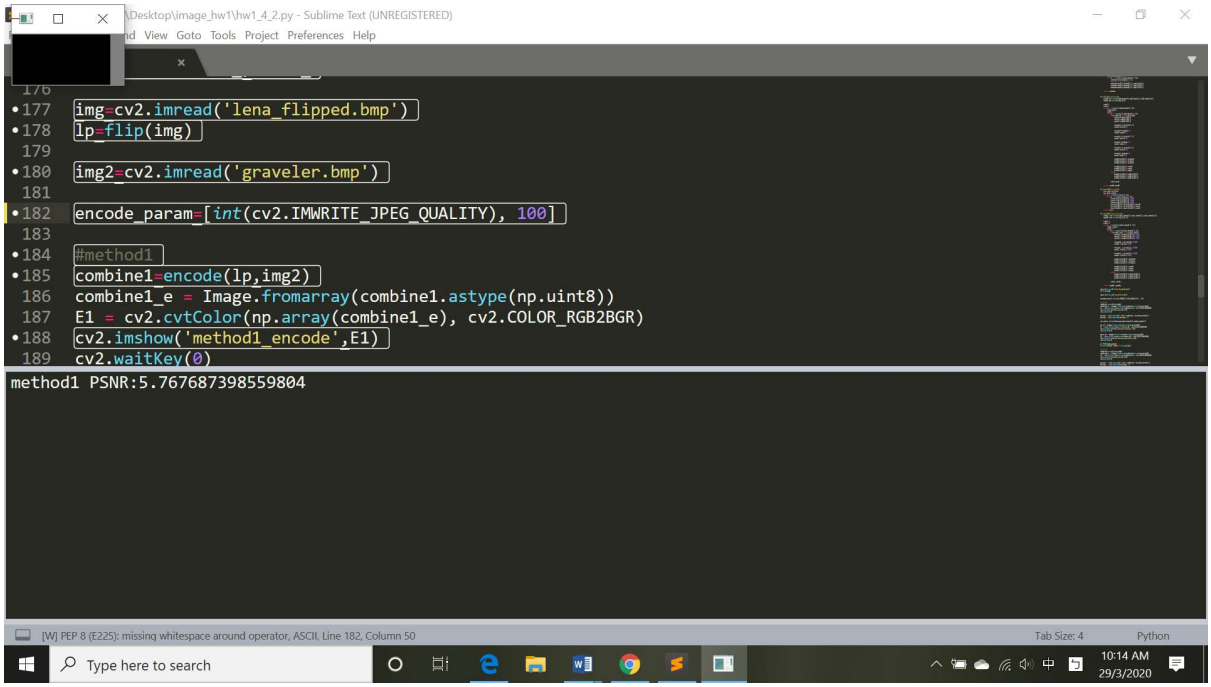

C:\Users\chrystal212\Desktop\image_hw1\hw1_4_2.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
hw1_4_2.py
img=cv2.imread('lena_flipped.bmp')
lp=flip(img)
img2=cv2.imread('graveler.bmp')
encode_param=[int(cv2.IMWRITE_JPEG_QUALITY), 100]
#method1
combine1=encode(lp,img2)
combine1_e = Image.fromarray(combine1.astype(np.uint8))
E1 = cv2.cvtColor(np.array(combine1_e), cv2.COLOR_RGB2BGR)
cv2.imshow('method1_encode',E1)
cv2.waitKey(0)
method1_decode_original
[W] PEP 8 (E225): missing whitespace around operator, ASCII, Line 182, Column 50
Tab Size: 4
Python
Type here to search
ENG INTL
10:04 AM
29/3/2020

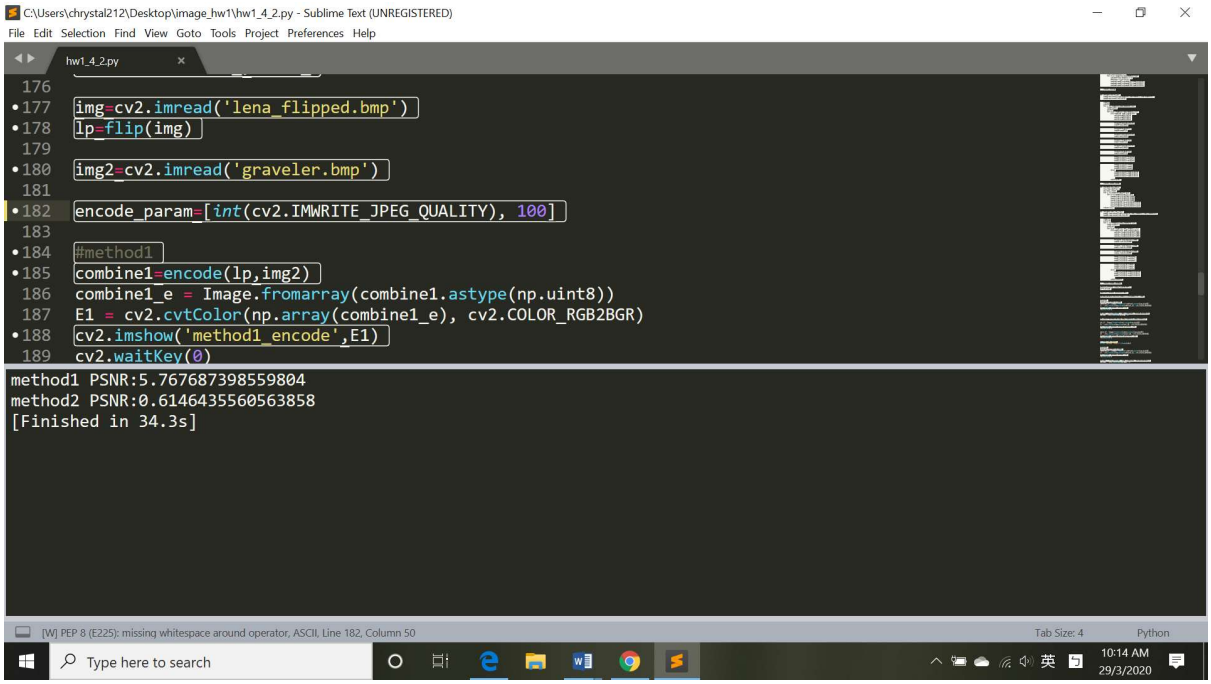

\Desktop\image_hw1\hw1_4_2.py - Sublime Text (UNREGISTERED)
View Goto Tools Project Preferences Help
img=cv2.imread('lena_flipped.bmp')
lp=flip(img)
img2=cv2.imread('graveler.bmp')
encode_param=[int(cv2.IMWRITE_JPEG_QUALITY), 100]
#method1
combine1=encode(lp,img2)
combine1_e = Image.fromarray(combine1.astype(np.uint8))
E1 = cv2.cvtColor(np.array(combine1_e), cv2.COLOR_RGB2BGR)
cv2.imshow('method1_encode',E1)
cv2.waitKey(0)
[W] PEP 8 (E225): missing whitespace around operator, ASCII, Line 182, Column 50
Tab Size: 4
Python
Type here to search
ENG INTL
10:04 AM
29/3/2020



最後，我們用 compression ratio 是 100 來做。

可以看到在壓縮完又還原後，我們去 retrieve watermark 及 lena image。我們看到 lena image 的解析度不太有變化。而 watermark 的圖則是比 90 時好一點，可以大概看得到原來的樣貌。而你所得到的 PSNR 值是 5.767 左右，比 compression rate 為 90 時高一點。





這邊，我另外做了法二的壓縮還原後 retrieve watermark 後的展現，可以看到其所呈現的是一片黑，而 PSNR 也是極低，只有 0.6 左右。顯示當初的法二，即使如果你不壓縮，其所呈現的效果比較好。但經過壓縮還原後，你 watermark 的資訊會完全丟失。