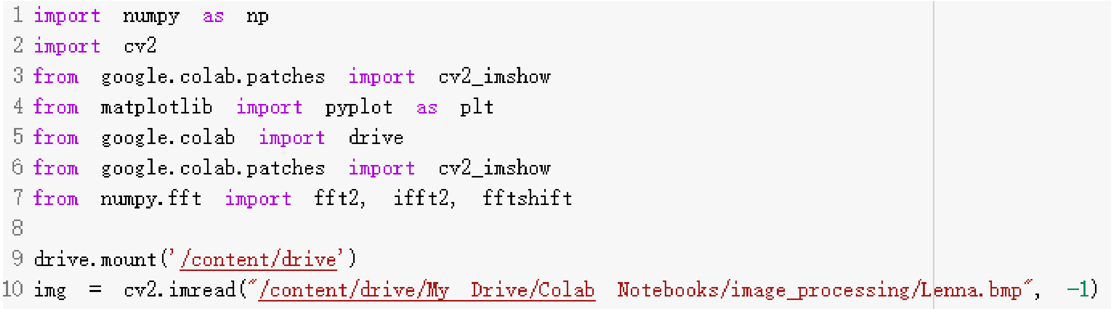
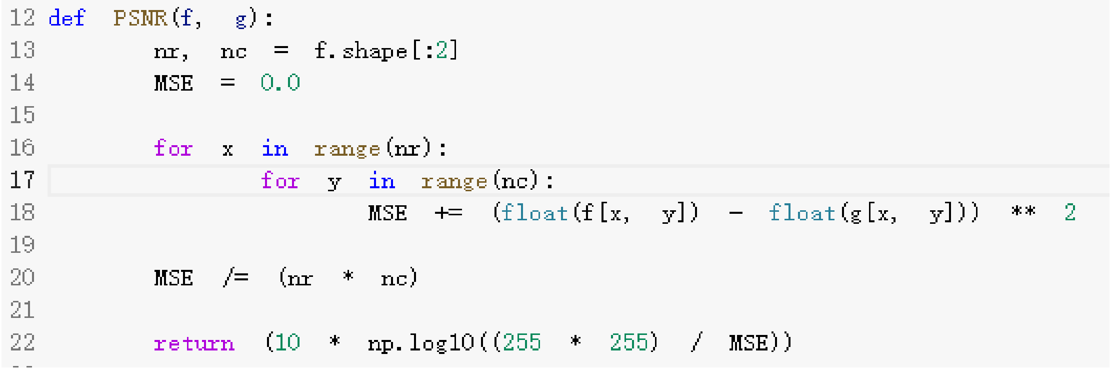
實戰三

讀取影像



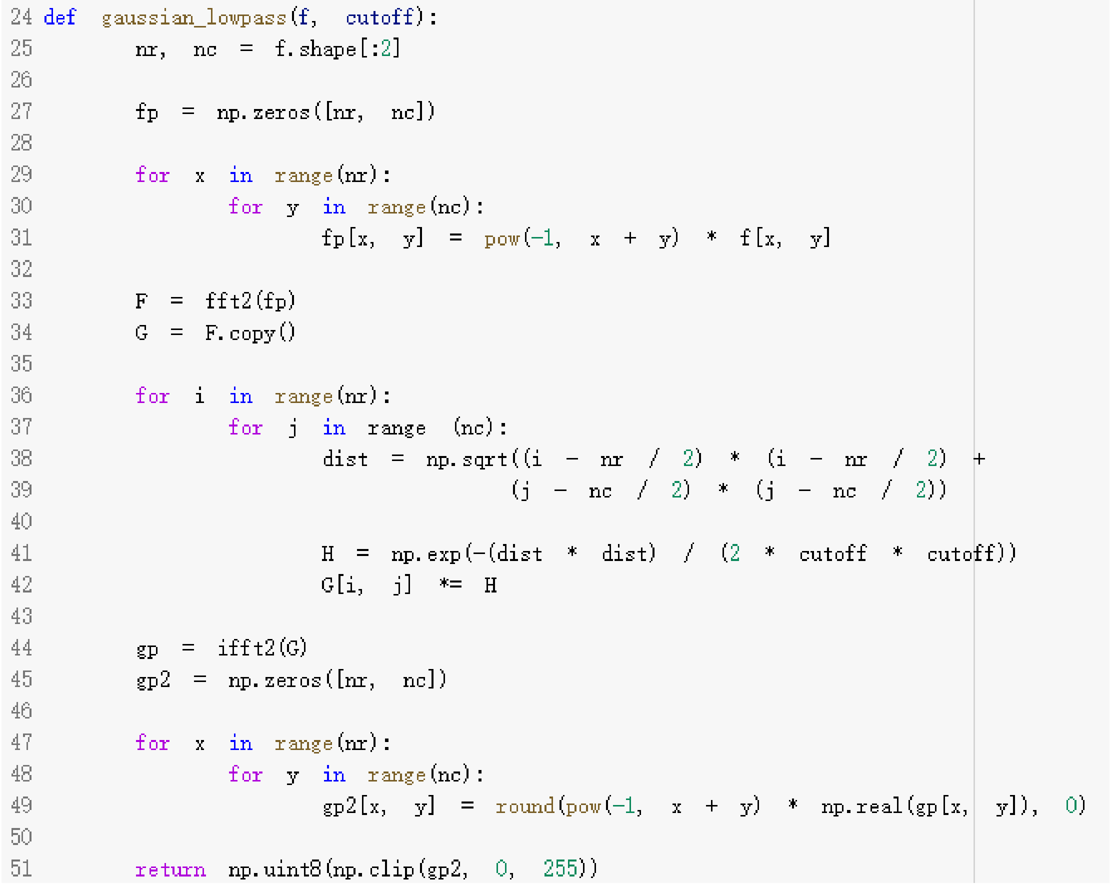
```python
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
from google.colab import drive
from google.colab.patches import cv2_imshow
from numpy.fft import fft2, ifft2, fftshift

drive.mount('/content/drive')
img = cv2.imread("/content/drive/My Drive/Colab Notebooks/image_processing/Lenna.bmp", -1)
```

PSNR 函數



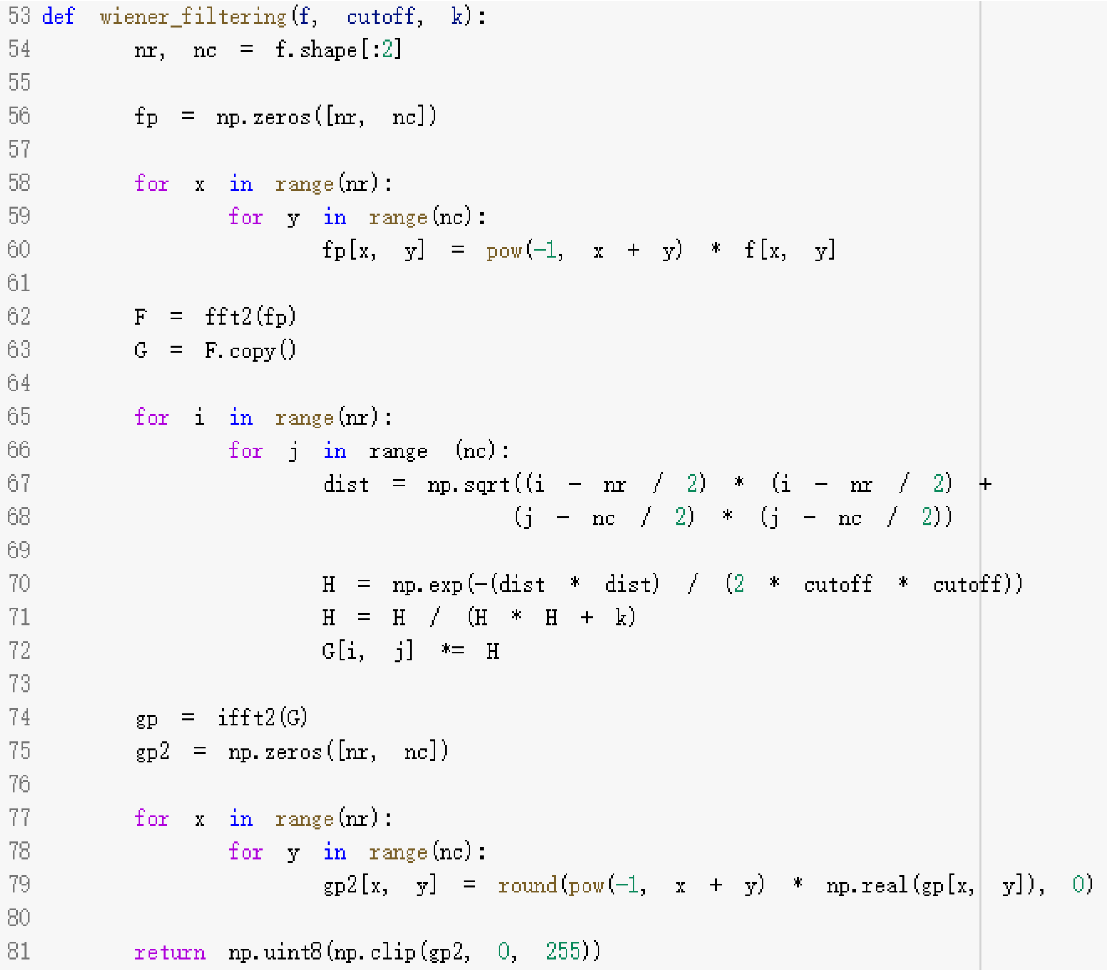
```python
def PSNR(f, g):
    nr, nc = f.shape[:2]
    MSE = 0.0

    for x in range(nr):
        for y in range(nc):
            MSE += (float(f[x, y]) - float(g[x, y])) ** 2

    MSE /= (nr * nc)

    return (10 * np.log10((255 * 255) / MSE))
```

影像失真函數(使用高斯函數模擬)



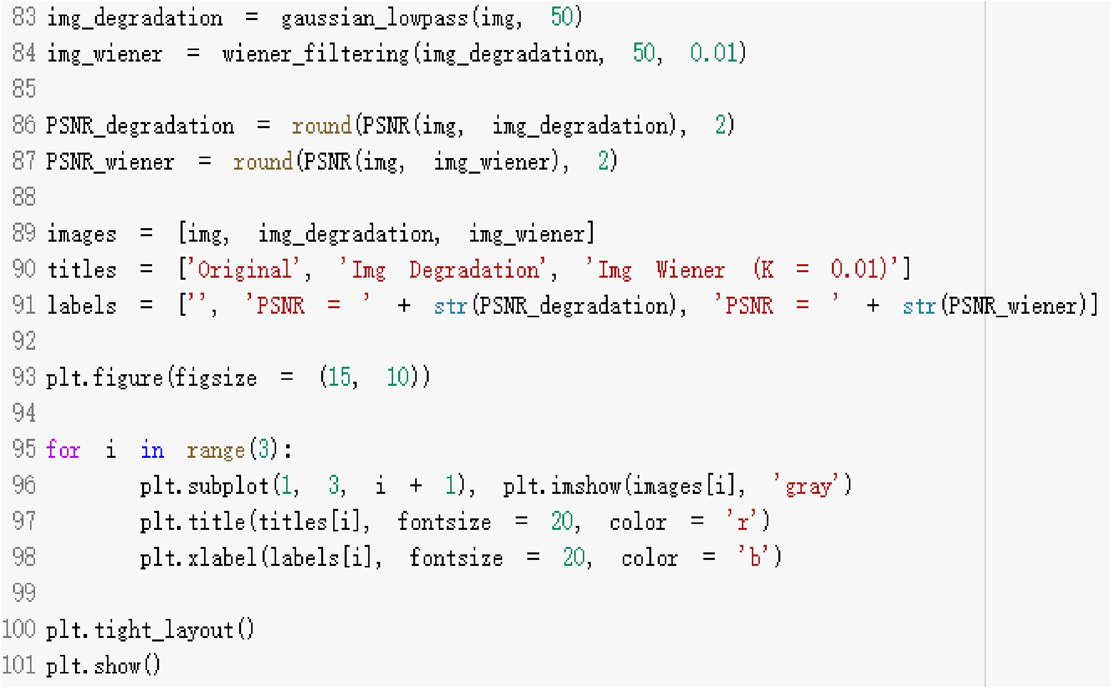
```python
def gaussian_lowpass(f, cutoff):
    nr, nc = f.shape[:2]

    fp = np.zeros([nr, nc])

    for x in range(nr):
        for y in range(nc):
            fp[x, y] = pow(-1, x + y) * f[x, y]

    F = fft2(fp)
    G = F.copy()

    for i in range(nr):
        for j in range (nc):
            dist = np.sqrt((i - nr / 2) * (i - nr / 2) +
                           (j - nc / 2) * (j - nc / 2))

            H = np.exp(-(dist * dist) / (2 * cutoff * cutoff))
            G[i, j] *= H

    gp = ifft2(G)
    gp2 = np.zeros([nr, nc])

    for x in range(nr):
        for y in range(nc):
            gp2[x, y] = round(pow(-1, x + y) * np.real(gp[x, y]), 0)

    return np.uint8(np.clip(gp2, 0, 255))
```

維納斯濾波函數



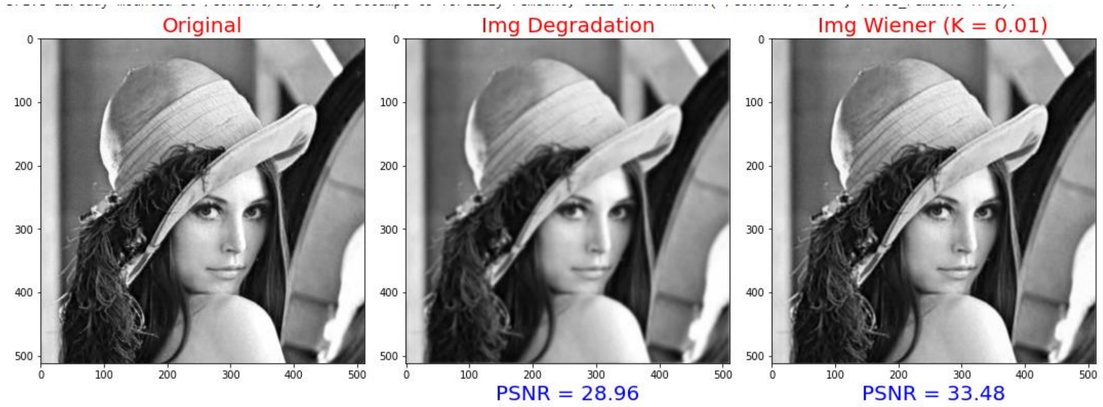
```python
def wiener_filtering(f, cutoff, k):
    nr, nc = f.shape[:2]

    fp = np.zeros([nr, nc])

    for x in range(nr):
        for y in range(nc):
            fp[x, y] = pow(-1, x + y) * f[x, y]

    F = fft2(fp)
    G = F.copy()

    for i in range(nr):
        for j in range (nc):
            dist = np.sqrt((i - nr / 2) * (i - nr / 2) +
                           (j - nc / 2) * (j - nc / 2))

            H = np.exp(-(dist * dist) / (2 * cutoff * cutoff))
            H = H / (H * H + k)
            G[i, j] *= H

    gp = ifft2(G)
    gp2 = np.zeros([nr, nc])

    for x in range(nr):
        for y in range(nc):
            gp2[x, y] = round(pow(-1, x + y) * np.real(gp[x, y]), 0)

    return np.uint8(np.clip(gp2, 0, 255))
```

結果輸出



```python
img_degradation = gaussian_lowpass(img, 50)
img_wiener = wiener_filtering(img_degradation, 50, 0.01)

PSNR_degradation = round(PSNR(img, img_degradation), 2)
PSNR_wiener = round(PSNR(img, img_wiener), 2)

images = [img, img_degradation, img_wiener]
titles = ['Original', 'Img Degradation', 'Img Wiener (K = 0.01)']
labels = ['', 'PSNR = ' + str(PSNR_degradation), 'PSNR = ' + str(PSNR_wiener)]

plt.figure(figsize = (15, 10))

for i in range(3):
    plt.subplot(1, 3, i + 1), plt.imshow(images[i], 'gray')
    plt.title(titles[i], fontsize = 20, color = 'r')
    plt.xlabel(labels[i], fontsize = 20, color = 'b')

plt.tight_layout()
plt.show()
```

結果

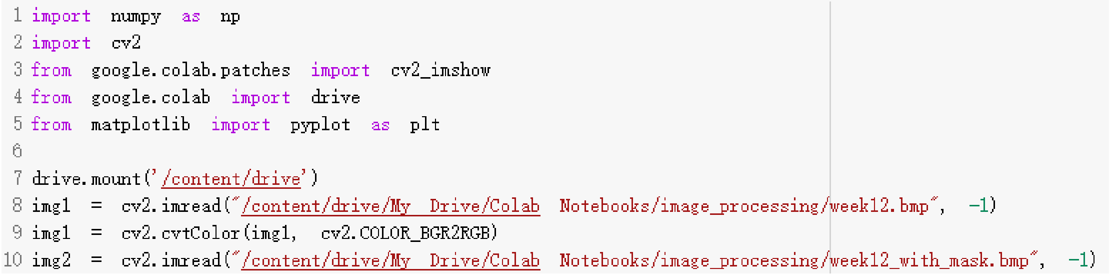


結論

單用肉眼來看的話，維納斯函數的影像還原效果相當不錯

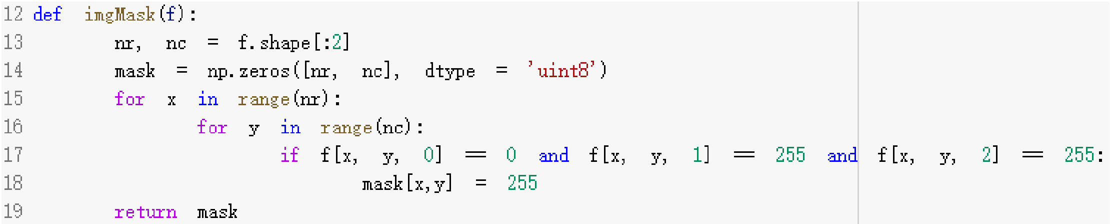
實戰四

讀取影像



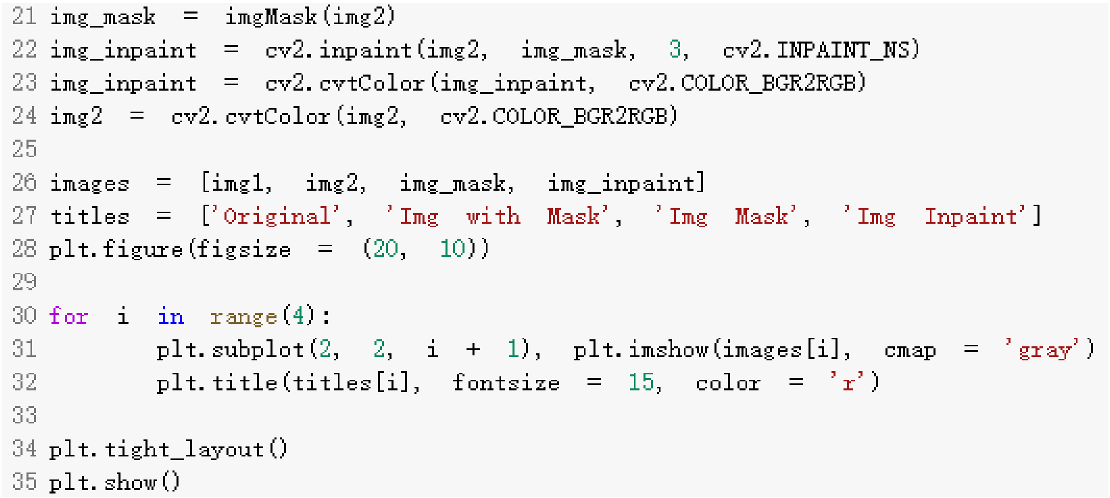
```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
from google.colab import drive
from matplotlib import pyplot as plt

drive.mount('/content/drive')
img1 = cv2.imread("/content/drive/My Drive/Colab Notebooks/image_processing/week12.bmp", -1)
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2 = cv2.imread("/content/drive/My Drive/Colab Notebooks/image_processing/week12_with_mask.bmp", -1)
```

影像補繪函數



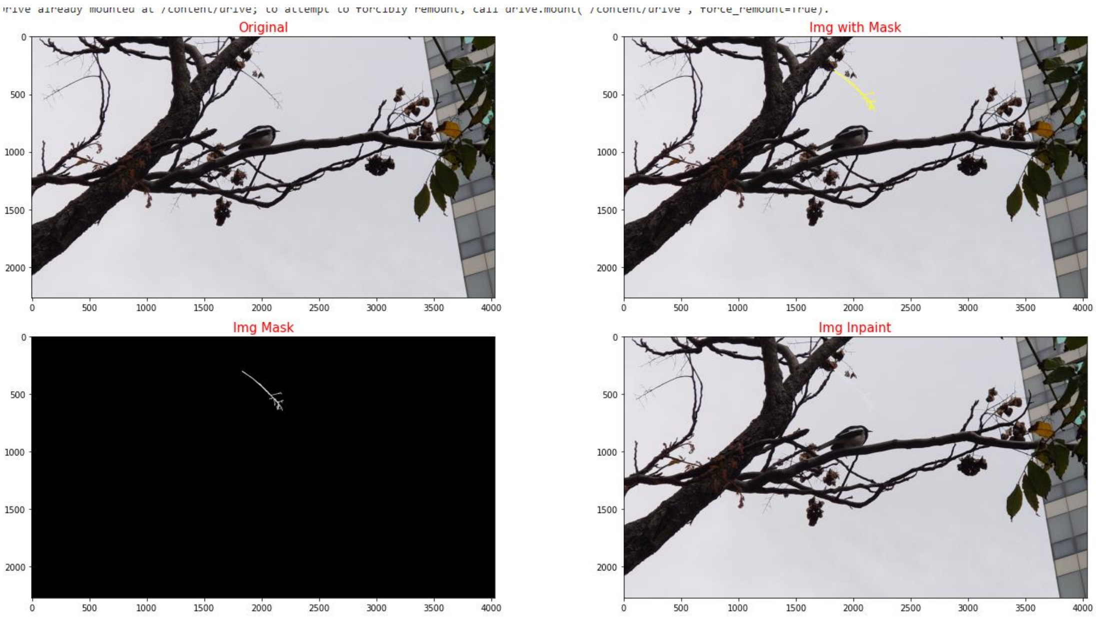
```
def imgMask(f):
    nr, nc = f.shape[:2]
    mask = np.zeros([nr, nc], dtype = 'uint8')
    for x in range(nr):
        for y in range(nc):
            if f[x, y, 0] == 0 and f[x, y, 1] == 255 and f[x, y, 2] == 255:
                mask[x,y] = 255
    return mask
```

結果輸出



```
img_mask = imgMask(img2)
img_inpaint = cv2.inpaint(img2, img_mask, 3, cv2.INPAINT_NS)
img_inpaint = cv2.cvtColor(img_inpaint, cv2.COLOR_BGR2RGB)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)

images = [img1, img2, img_mask, img_inpaint]
titles = ['Original', 'Img with Mask', 'Img Mask', 'Img Inpaint']
plt.figure(figsize = (20, 10))

for i in range(4):
    plt.subplot(2, 2, i + 1), plt.imshow(images[i], cmap = 'gray')
    plt.title(titles[i], fontsize = 15, color = 'r')

plt.tight_layout()
plt.show()
```

結果



```
rive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

結論

由結果可以看出，python 提供的補繪函數不太完美，還是可以看出些許端倪

問答題

(1) 試說明影像雜訊的分析方法
影像雜訊的分析方法主要分為總誤差、均分誤差、均分根誤差、峰值雜訊比，主要用於理解雜訊的分布與本質。

(2) 試解釋反濾波的技術挑戰
因自然環境中的影像通常含有雜訊，而我們無法事先得知此雜訊函數 N(u,v) 為何，也因此想要實現理想的反濾波技術是有困難的。

(3) 試說明影像補繪技術的使用時機
若有不想要出現的東西出現在影像中，可運用影像補繪技術，將該東西從影像中剃除。

(4) 請簡單說明 CH07 影像還原技術的課程重點 (不超過 100 字)
去了解影像還原技術有哪些，及如何實踐，並運用雜訊分析函數讓我們看出各影像還原技術成效大概為何。