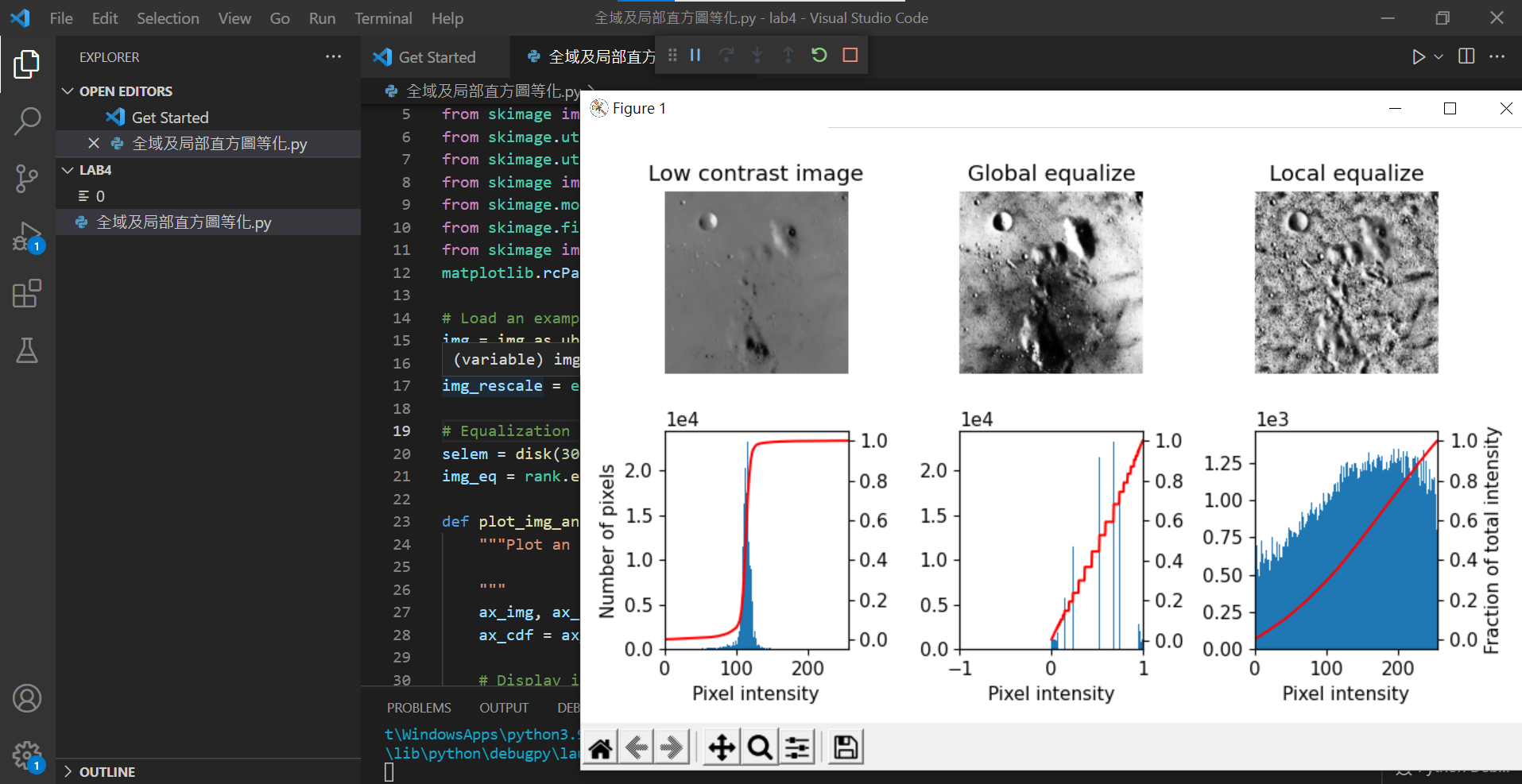
1)使用scikit-image完成全域直方圖等化



中間的圖為全域直方圖等化

Code:

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

from skimage import data

from skimage.util.dtype import dtype\_range

from skimage.util import img\_as\_ubyte

from skimage import exposure

from skimage.morphology import disk

from skimage.filters import rank

from skimage import io

matplotlib.rcParams['font.size'] = 11

# Load an example image

img = img\_as\_ubyte(data.moon())

# Global equalize

img\_rescale = exposure.equalize\_hist(img)

# Equalization

selem = disk(30)

img\_eq = rank.equalize(img, selem=selem)

def plot\_img\_and\_hist(image, axes, bins=256):

    """Plot an image along with its histogram and cumulative histogram.

    """

    ax\_img, ax\_hist = axes

    ax\_cdf = ax\_hist.twinx()

    # Display image

    ax\_img.imshow(image, cmap=plt.cm.gray)

    ax\_img.set\_axis\_off()

    # Display histogram

    ax\_hist.hist(image.ravel(), bins=bins)

    ax\_hist.ticklabel\_format(axis='y', style='scientific', scilimits=(0, 0))

    ax\_hist.set\_xlabel('Pixel intensity')

    xmin, xmax = dtype\_range[image.dtype.type]

    ax\_hist.set\_xlim(xmin, xmax)

    # Display cumulative distribution

    img\_cdf, bins = exposure.cumulative\_distribution(image, bins)

    ax\_cdf.plot(bins, img\_cdf, 'r')

    return ax\_img, ax\_hist, ax\_cdf

# Display results

fig = plt.figure(figsize=(8, 5))

axes = np.zeros((2, 3), dtype=object)

axes[0, 0] = plt.subplot(2, 3, 1)

axes[0, 1] = plt.subplot(2, 3, 2, sharex=axes[0, 0], sharey=axes[0, 0])

axes[0, 2] = plt.subplot(2, 3, 3, sharex=axes[0, 0], sharey=axes[0, 0])

axes[1, 0] = plt.subplot(2, 3, 4)

axes[1, 1] = plt.subplot(2, 3, 5)

axes[1, 2] = plt.subplot(2, 3, 6)

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img, axes[:, 0])

ax\_img.set\_title('Low contrast image')

ax\_hist.set\_ylabel('Number of pixels')

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img\_rescale, axes[:, 1])

ax\_img.set\_title('Global equalize')

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img\_eq, axes[:, 2])

ax\_img.set\_title('Local equalize')

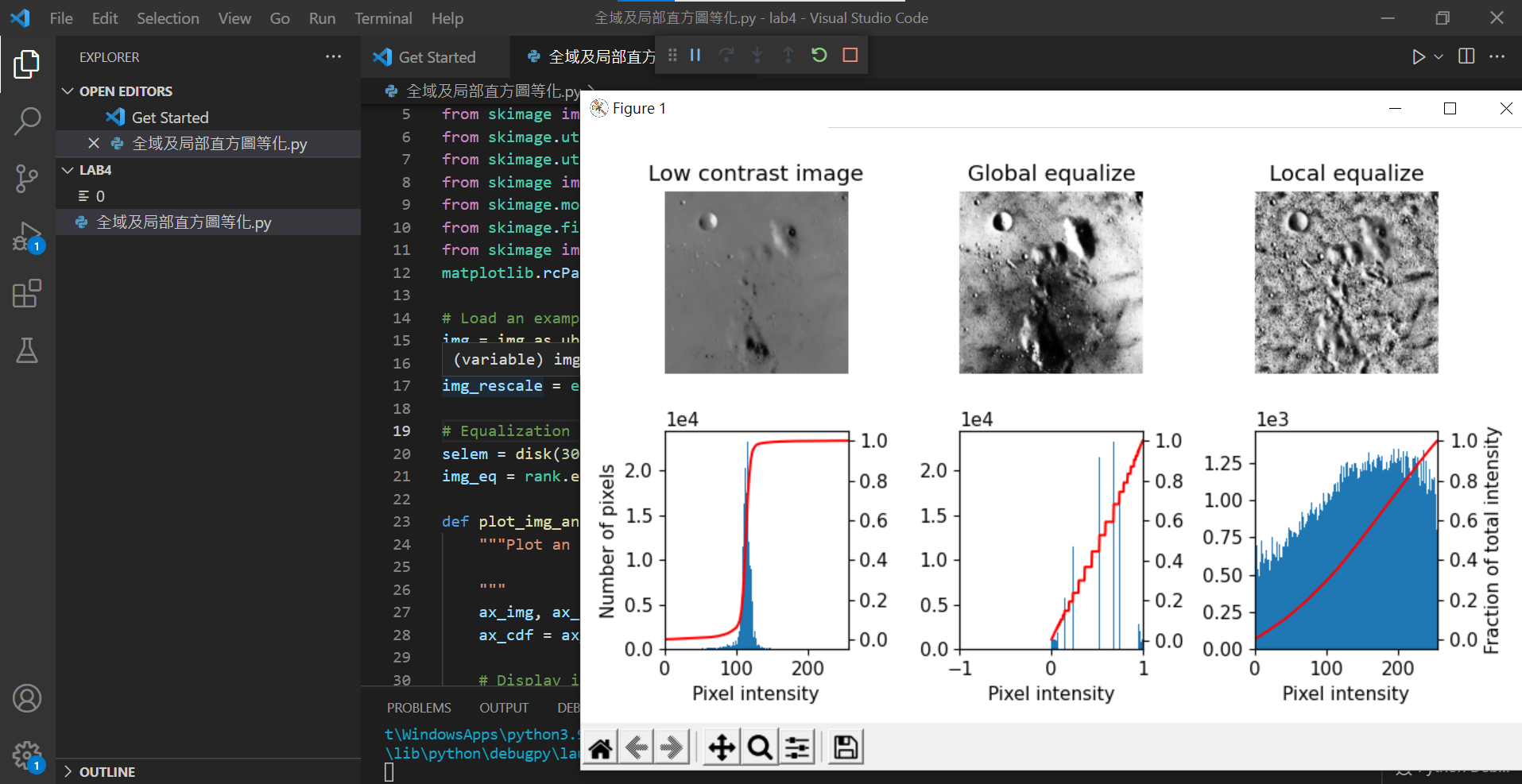
ax\_cdf.set\_ylabel('Fraction of total intensity')

# prevent overlap of y-axis labels

fig.tight\_layout()

plt.show()

2)使用scikit-image完成局部直方圖等化



右邊的圖為局部直方圖等化

Code:

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

from skimage import data

from skimage.util.dtype import dtype\_range

from skimage.util import img\_as\_ubyte

from skimage import exposure

from skimage.morphology import disk

from skimage.filters import rank

from skimage import io

matplotlib.rcParams['font.size'] = 11

# Load an example image

img = img\_as\_ubyte(data.moon())

# Global equalize

img\_rescale = exposure.equalize\_hist(img)

# Equalization

selem = disk(30)

img\_eq = rank.equalize(img, selem=selem)

def plot\_img\_and\_hist(image, axes, bins=256):

    """Plot an image along with its histogram and cumulative histogram.

    """

    ax\_img, ax\_hist = axes

    ax\_cdf = ax\_hist.twinx()

    # Display image

    ax\_img.imshow(image, cmap=plt.cm.gray)

    ax\_img.set\_axis\_off()

    # Display histogram

    ax\_hist.hist(image.ravel(), bins=bins)

    ax\_hist.ticklabel\_format(axis='y', style='scientific', scilimits=(0, 0))

    ax\_hist.set\_xlabel('Pixel intensity')

    xmin, xmax = dtype\_range[image.dtype.type]

    ax\_hist.set\_xlim(xmin, xmax)

    # Display cumulative distribution

    img\_cdf, bins = exposure.cumulative\_distribution(image, bins)

    ax\_cdf.plot(bins, img\_cdf, 'r')

    return ax\_img, ax\_hist, ax\_cdf

# Display results

fig = plt.figure(figsize=(8, 5))

axes = np.zeros((2, 3), dtype=object)

axes[0, 0] = plt.subplot(2, 3, 1)

axes[0, 1] = plt.subplot(2, 3, 2, sharex=axes[0, 0], sharey=axes[0, 0])

axes[0, 2] = plt.subplot(2, 3, 3, sharex=axes[0, 0], sharey=axes[0, 0])

axes[1, 0] = plt.subplot(2, 3, 4)

axes[1, 1] = plt.subplot(2, 3, 5)

axes[1, 2] = plt.subplot(2, 3, 6)

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img, axes[:, 0])

ax\_img.set\_title('Low contrast image')

ax\_hist.set\_ylabel('Number of pixels')

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img\_rescale, axes[:, 1])

ax\_img.set\_title('Global equalize')

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img\_eq, axes[:, 2])

ax\_img.set\_title('Local equalize')

ax\_cdf.set\_ylabel('Fraction of total intensity')

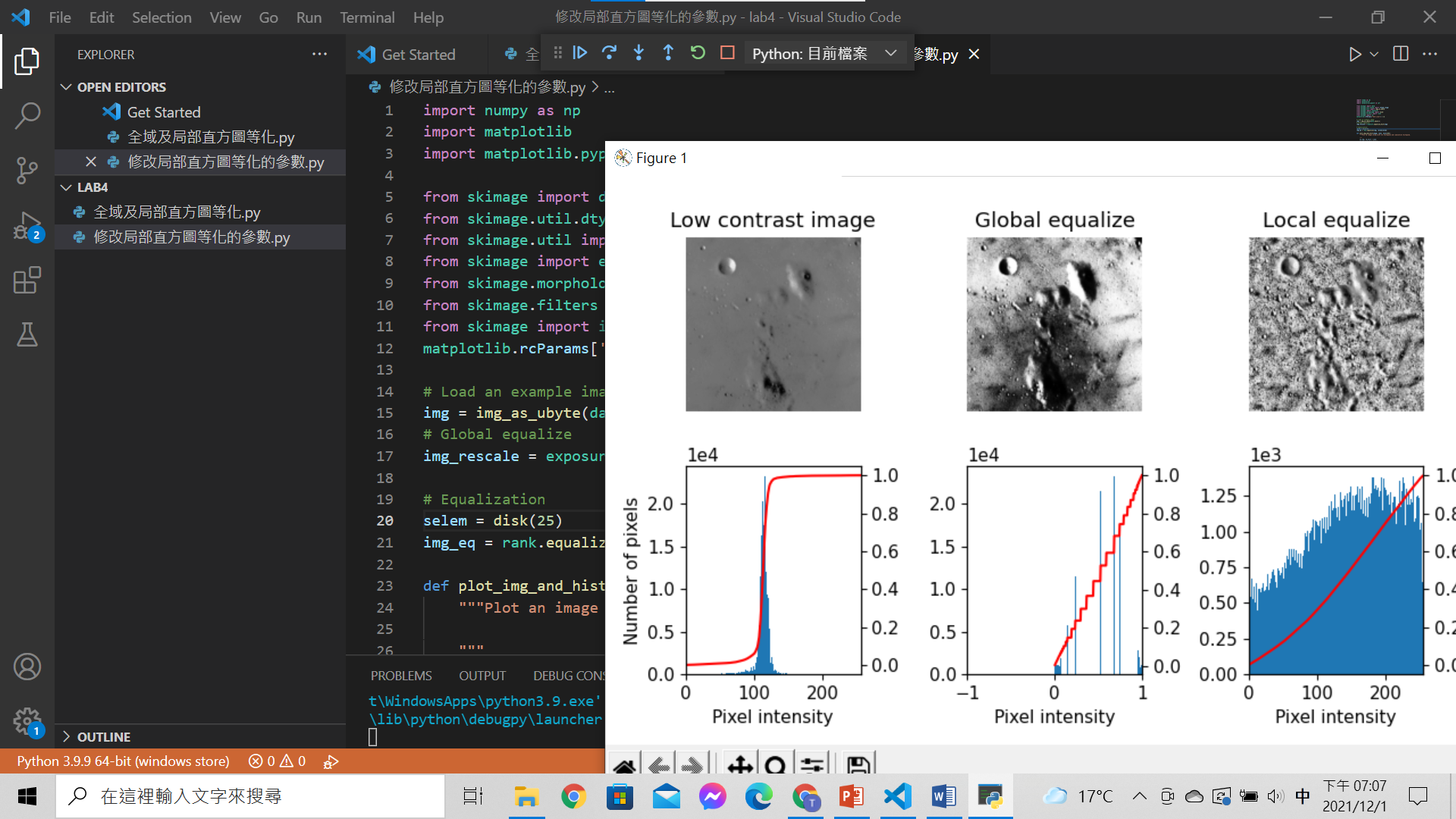
# prevent overlap of y-axis labels

fig.tight\_layout()

plt.show()

3)修改局部直方圖等化的參數，找出最佳的結果

經過多次測試後發現參數25能產生最佳的結果



Code:

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

from skimage import data

from skimage.util.dtype import dtype\_range

from skimage.util import img\_as\_ubyte

from skimage import exposure

from skimage.morphology import disk

from skimage.filters import rank

from skimage import io

matplotlib.rcParams['font.size'] = 11

# Load an example image

img = img\_as\_ubyte(data.moon())

# Global equalize

img\_rescale = exposure.equalize\_hist(img)

# Equalization

selem = disk(25)

img\_eq = rank.equalize(img, selem=selem)

def plot\_img\_and\_hist(image, axes, bins=256):

    """Plot an image along with its histogram and cumulative histogram.

    """

    ax\_img, ax\_hist = axes

    ax\_cdf = ax\_hist.twinx()

    # Display image

    ax\_img.imshow(image, cmap=plt.cm.gray)

    ax\_img.set\_axis\_off()

    # Display histogram

    ax\_hist.hist(image.ravel(), bins=bins)

    ax\_hist.ticklabel\_format(axis='y', style='scientific', scilimits=(0, 0))

    ax\_hist.set\_xlabel('Pixel intensity')

    xmin, xmax = dtype\_range[image.dtype.type]

    ax\_hist.set\_xlim(xmin, xmax)

    # Display cumulative distribution

    img\_cdf, bins = exposure.cumulative\_distribution(image, bins)

    ax\_cdf.plot(bins, img\_cdf, 'r')

    return ax\_img, ax\_hist, ax\_cdf

# Display results

fig = plt.figure(figsize=(8, 5))

axes = np.zeros((2, 3), dtype=object)

axes[0, 0] = plt.subplot(2, 3, 1)

axes[0, 1] = plt.subplot(2, 3, 2, sharex=axes[0, 0], sharey=axes[0, 0])

axes[0, 2] = plt.subplot(2, 3, 3, sharex=axes[0, 0], sharey=axes[0, 0])

axes[1, 0] = plt.subplot(2, 3, 4)

axes[1, 1] = plt.subplot(2, 3, 5)

axes[1, 2] = plt.subplot(2, 3, 6)

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img, axes[:, 0])

ax\_img.set\_title('Low contrast image')

ax\_hist.set\_ylabel('Number of pixels')

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img\_rescale, axes[:, 1])

ax\_img.set\_title('Global equalize')

ax\_img, ax\_hist, ax\_cdf = plot\_img\_and\_hist(img\_eq, axes[:, 2])

ax\_img.set\_title('Local equalize')

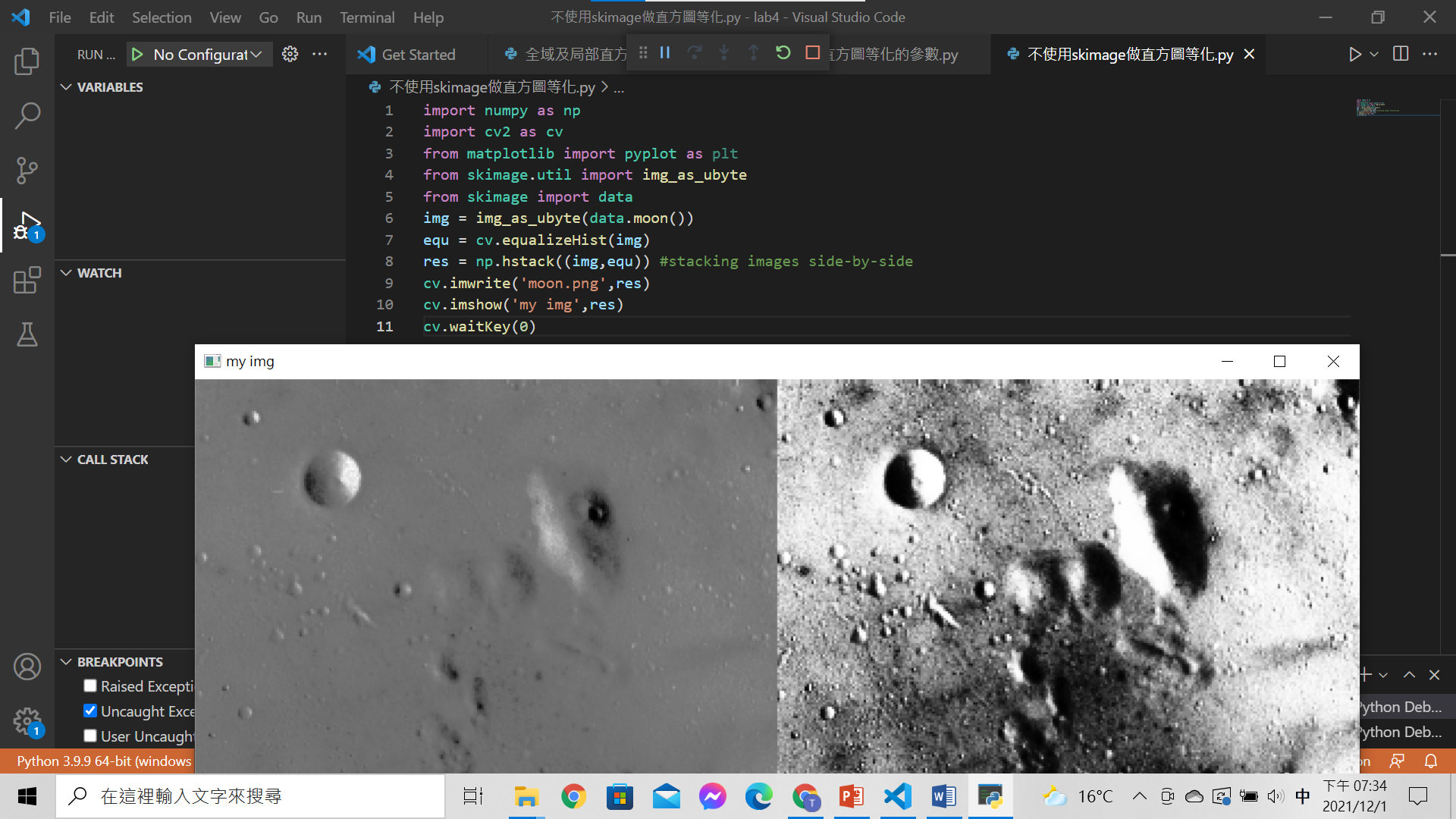
ax\_cdf.set\_ylabel('Fraction of total intensity')

# prevent overlap of y-axis labels

fig.tight\_layout()

plt.show()

4)不使用scikit-image完成全域直方圖等化



Code:

import numpy as np

import cv2 as cv

from matplotlib import pyplot as plt

from skimage.util import img\_as\_ubyte

from skimage import data

img = img\_as\_ubyte(data.moon())

equ = cv.equalizeHist(img)

res = np.hstack((img,equ)) #stacking images side-by-side

cv.imwrite('moon.png',res)

cv.imshow('my img',res)

cv.waitKey(0)

心得

這次的實習活動除了讓我複習直方圖等化的基本原理也讓我對全域直方圖等化及局部直方圖等化有更進一步的了解。雖然範例程式中還是有一些模組式我不熟悉的，但是透過網路的學習及透過實作結果的猜測，也漸漸的對這些模組越來越熟悉，未來期許能深入的了解這些模組是如何運作的，以便能更方便的應用他們。