**01 Matlab Code**

1. **Scale Transformation Function**

설명 : 스케일 변환 함수

% Scaling transformation function

function y = Scale(x, max\_value, min\_value)

y = (255/(max\_value-min\_value)\*(x-min\_value));

end

1. **Equalize Transformation Function**

설명 : 평활화 변환 함수

%Equalize Transform Fuction, x is histogram value

function y = ETF(x)

m = size(x);

sum = 0;

for i = 1:m

sum = sum+x(i);

y(i,1) = sum;

end

end

1. **Image Equalizing Fuction**

설명 : 이미지가 ETF에 의해 평활화 되도록 도와주는 함수

% x1 is original image and x2 is ETF

function y = EF(x1, x2)

[m, n] = size(x1);

for i = 1:m

for j = 1:n

y(i, j) = x2(x1(i, j)+1);

end

end

1. **Main Code**

%%% Robot Vision%%%

%%% Dept. of Electronic Engineering

%%% 201314651 Lee Wonjai

IM\_Pollen = imread('C:\Users\user\OneDrive\¹ÙÅÁ È­¸é\4ÇÐ³â 2ÇÐ±â\·Îº¿ºñÀü\Original Images\dipum\_images\_ch03\Fig0308(a)(pollen).tif'); % read the targeted image

IM\_Moon = imread('C:\Users\user\OneDrive\¹ÙÅÁ È­¸é\4ÇÐ³â 2ÇÐ±â\·Îº¿ºñÀü\Original Images\dipum\_images\_ch03\Fig0310(a)(Moon Phobos).tif');

P\_Pollen = imhist(IM\_Pollen);

P\_Moon = imhist(IM\_Moon);

%Find Max and Min value of Pollen Image

Min1 = min(find(P\_Pollen))-1; %also can be done using min and max value of IM\_Pollen

Max1 = max(find(P\_Pollen))-1; %the reason of -1 is that array in matlab start to calculate the index of array as 1

%Find Max and Min value of Moon Image

Min2 = min(find(P\_Moon))-1;

Max2 = max(find(P\_Moon))-1;

IMScale\_Pollen = Scale(IM\_Pollen, Max1, Min1);

IMScale\_Moon = Scale(IM\_Moon, Max2, Min2);

PScale\_Pollen = imhist(IMScale\_Pollen);

PScale\_Moon = imhist(IMScale\_Moon);

[m1, n1] = size(IM\_Pollen);

[m2, n2] = size(IM\_Moon);

ETF\_Pollen = ETF(P\_Pollen)./(m1.\*n1).\*255;

ETF\_Moon = ETF(P\_Moon)./(m2.\*n2).\*255;

EF\_Pollen = uint8(EF(IM\_Pollen, ETF\_Pollen));

EF\_Moon = uint8(EF(IM\_Moon, ETF\_Moon));

CEF\_Pollen = histeq(IM\_Pollen, 256);

CEF\_Moon = histeq(IM\_Moon, 256);

% Compare the histogram of Original Image and scale transformed Image

figure(1);

subplot(1,2,1);

imhist(IM\_Pollen)

title('Original Image of Pollen')

subplot(1,2,2);

imhist(IMScale\_Pollen)

title('Scale transformed Image of Pollen')

figure(2);

subplot(1,2,1);

imhist(IM\_Moon)

title('Original Image of the Moon')

subplot(1,2,2);

imhist(IMScale\_Moon)

title('Scale transformed Image of the Moon')

% Compare the Original Image and scale transformed Image

figure(3);

subplot(1,2,1);

imshow(IM\_Pollen)

title('Original Image of Pollen')

subplot(1,2,2);

imshow(IMScale\_Pollen)

title('Scale transformed Image of Pollen')

figure(4)

subplot(1,2,1);

imshow(IM\_Moon)

title('Original Image of the Moon')

subplot(1,2,2);

imshow(IMScale\_Moon)

title('Scale transformed Image of the Moon')

% Compare the histogram of Original Image and equalized Image

figure(5);

subplot(1,2,1);

imhist(IM\_Pollen)

title('Original Image of Pollen')

subplot(1,2,2);

imhist(EF\_Pollen)

title('Equalized Image of Pollen')

figure(6);

subplot(1,2,1);

imhist(IM\_Moon)

title('Original Image of the Moon')

subplot(1,2,2);

imhist(EF\_Moon)

title('Equalized Image of the Moon')

% Compare the Original Image and equalized Image

figure(7);

subplot(1,2,1);

imshow(IM\_Pollen)

title('Original Image of Pollen')

subplot(1,2,2);

imshow(EF\_Pollen)

title('Equalized Image of Pollen')

figure(8);

subplot(1,2,1);

imshow(IM\_Moon)

title('Original Image of the Moon')

subplot(1,2,2);

imshow(EF\_Moon)

title('Equalized Image of the Moon')

%Compare whether equalization that I made is correct or not

figure(9)

subplot(1,2,1);

imhist(EF\_Pollen)

title('Equalization Fuction I made')

subplot(1,2,2);

imhist(CEF\_Pollen)

title('Equalization Fuction')

figure(10)

subplot(1,2,1);

imhist(EF\_Moon)

title('Equalization Fuction I made')

subplot(1,2,2);

imhist(CEF\_Moon)

title('Equalization Fuction')

figure(11)

subplot(1,2,1);

imshow(EF\_Pollen)

title('Equalization Fuction I made')

subplot(1,2,2);

imshow(CEF\_Pollen)

title('Equalization Fuction')

figure(12)

subplot(1,2,1);

imshow(EF\_Moon)

title('Equalization Fuction I made')

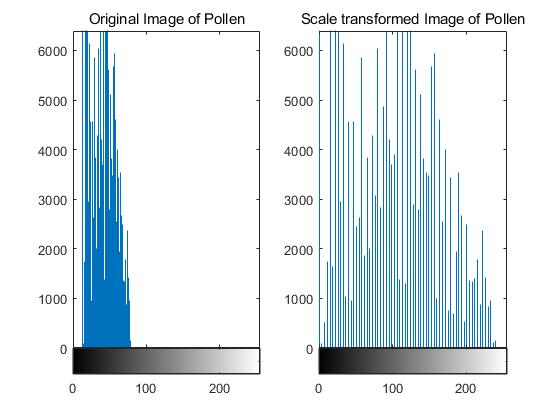
subplot(1,2,2);

imshow(CEF\_Moon)

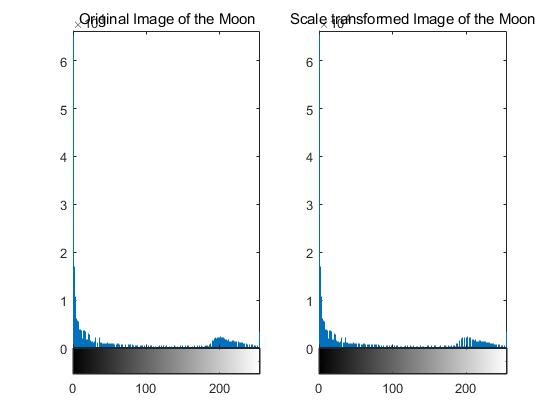
title('Equalization Fuction')

**02 Result 1**

1. **Histogram of Original and Scaled Image of Pollen**

****

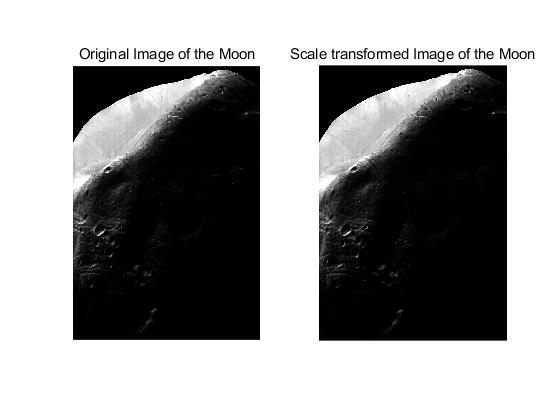
1. **Histogram of Original and Scaled Image of the Moon**



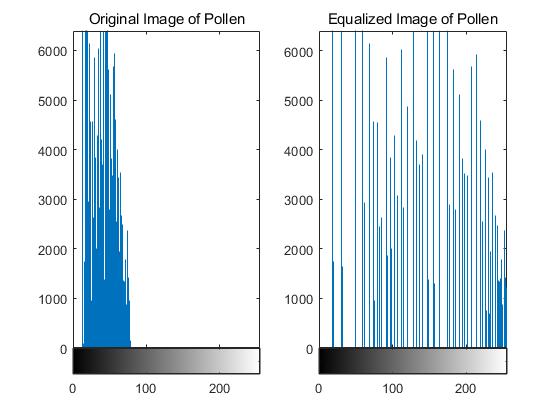
1. **Original and Scaled Image of Pollen**



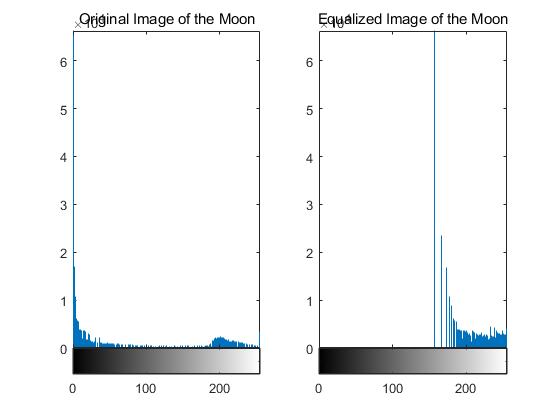
1. **Original and Scaled Image of the Moon**



1. **Histogram of Original and Equalized Image of Pollen**

****

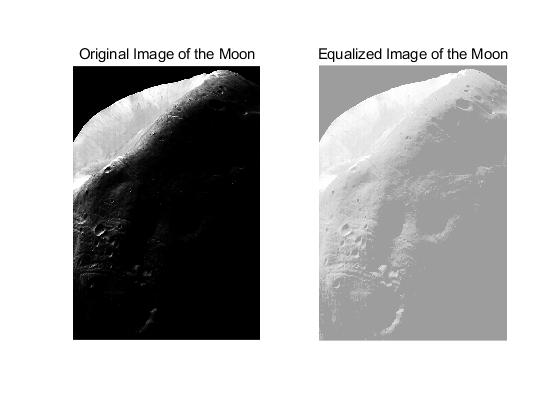
1. **Histogram of Original and Equalized Image of the Moon**



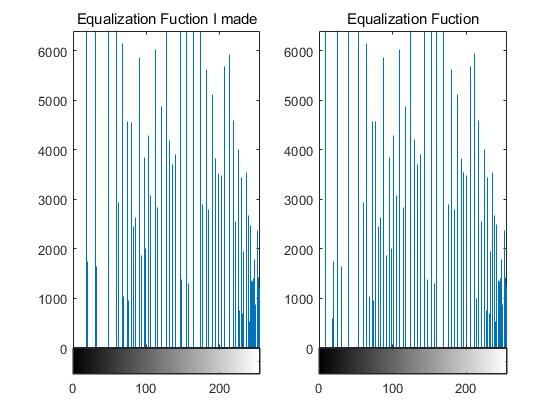
1. **Original and Equalized Image of Pollen**



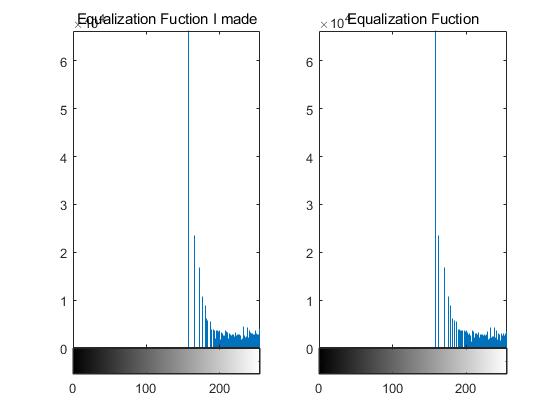
1. **Original and Equalized Image of the Moon**



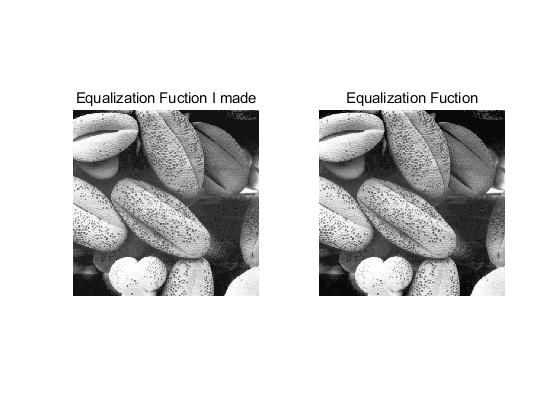
1. **Check the Function that I made whether right or not (Pollen)**



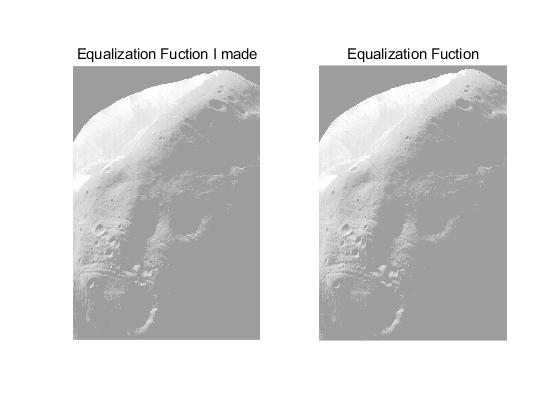
1. **Check the Function that I made whether is right or not (Moon)**

****

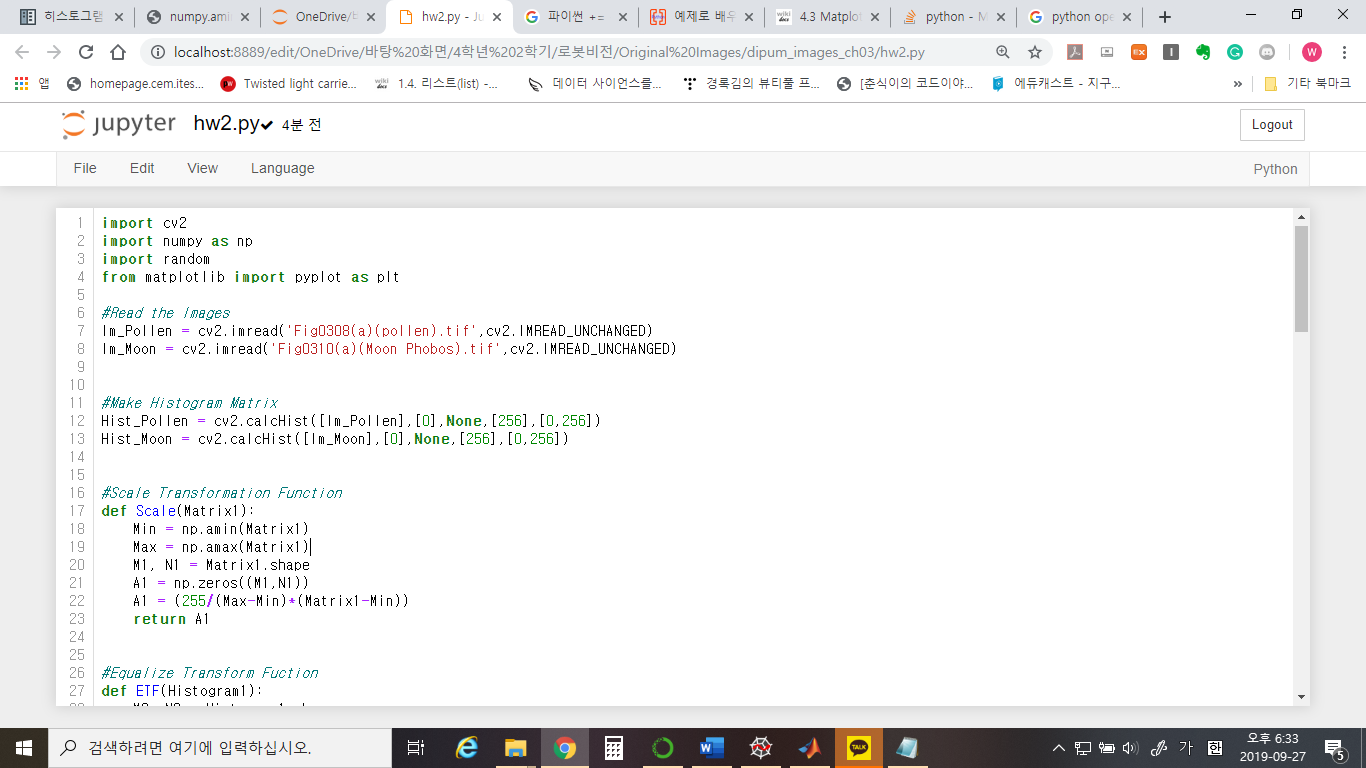
1. **Check the Function that I made whether right or not (Pollen)**

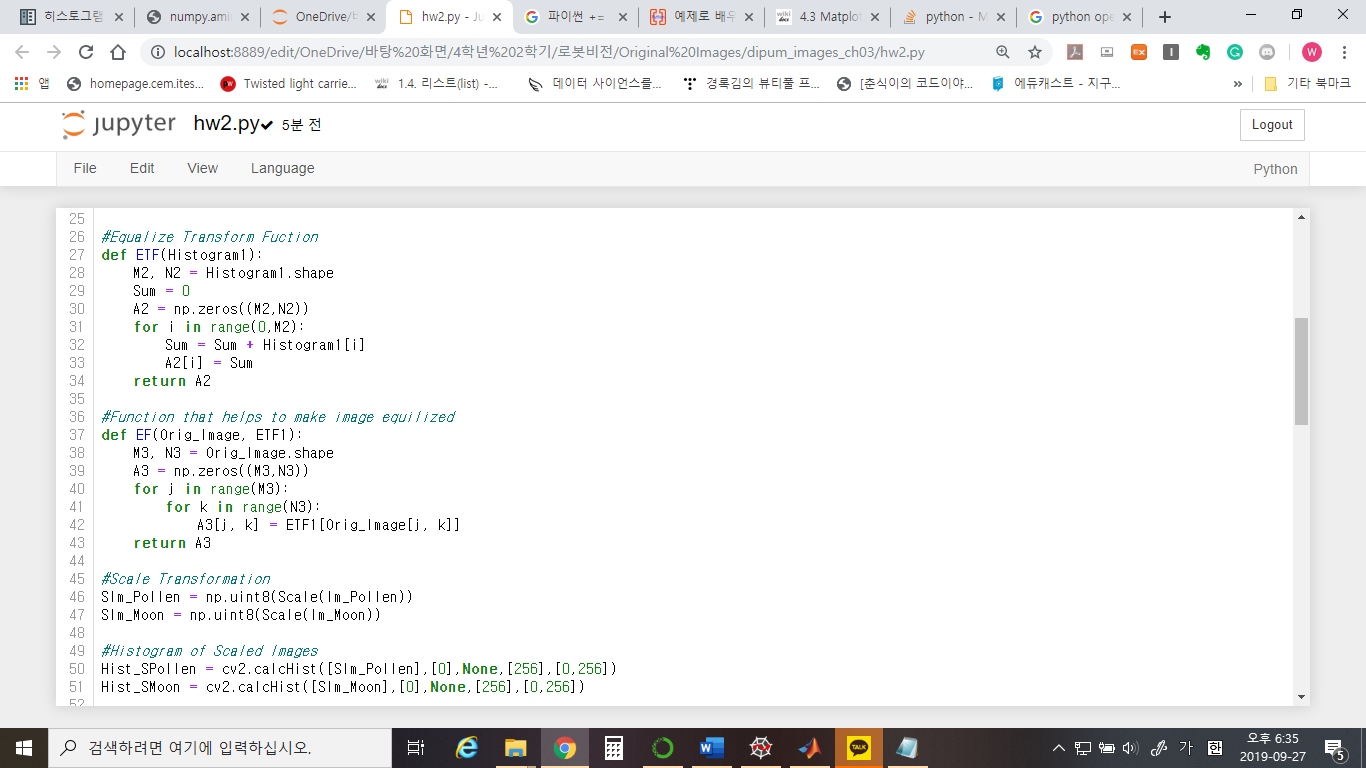


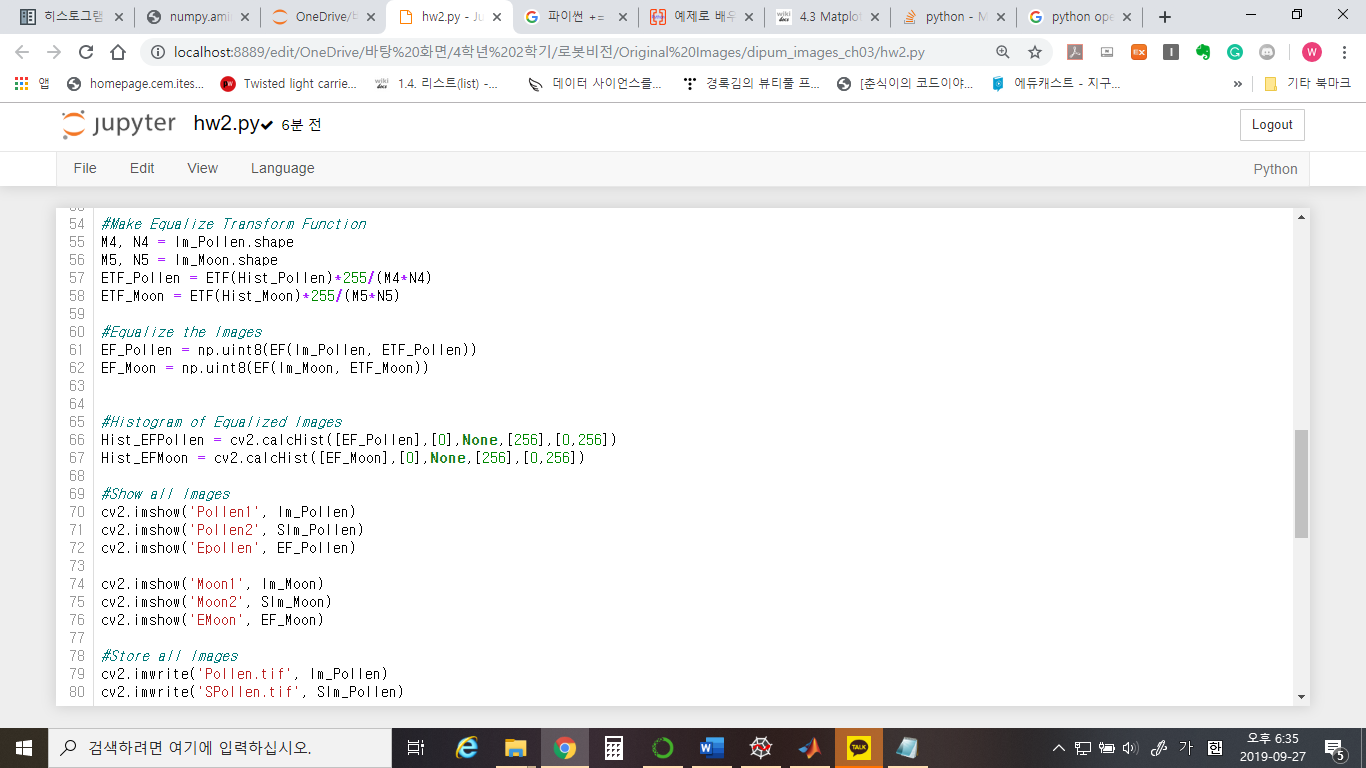
1. **Check the Function that I made whether right or not (Moon)**

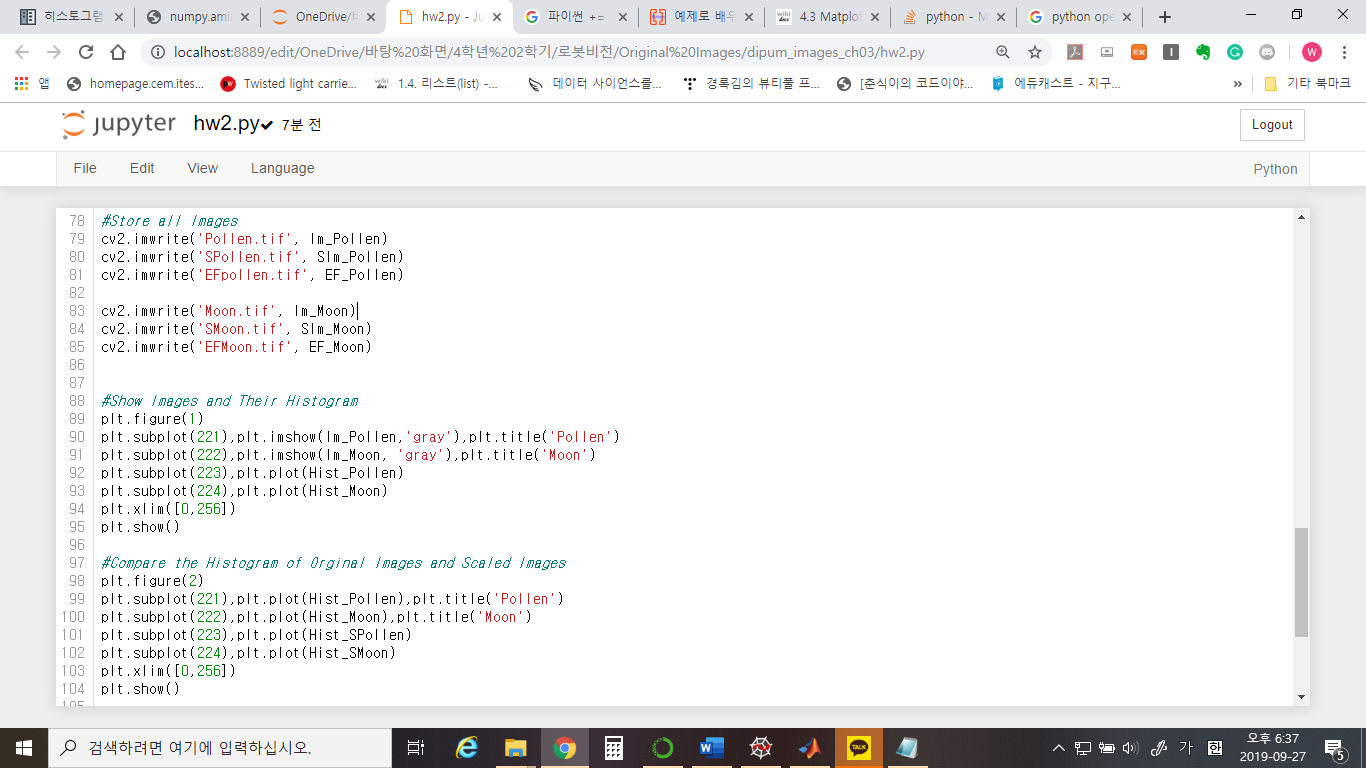


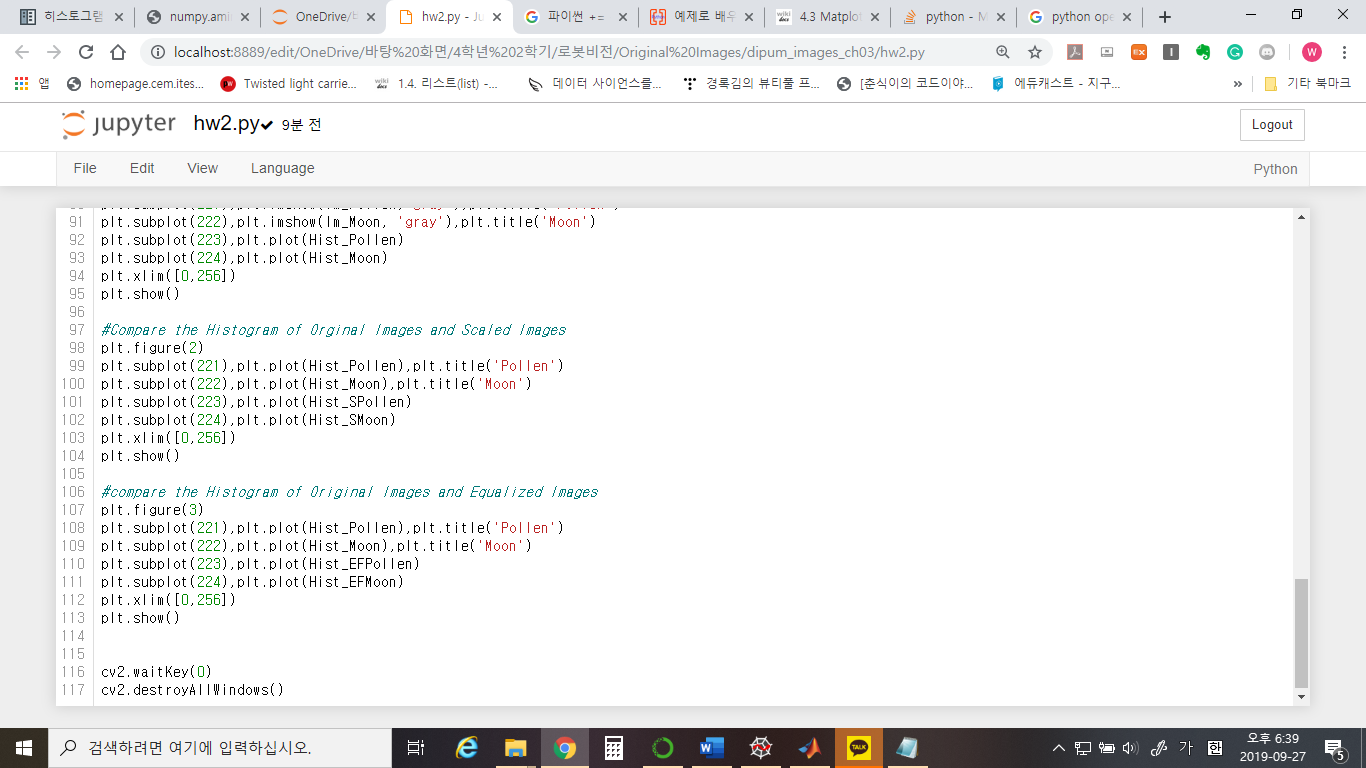
**02 심화학습(python 구현)**











**Result 2**

1. **Original and Scaled Image of Pollen**



1. **Original and Scaled Image of the Moon**



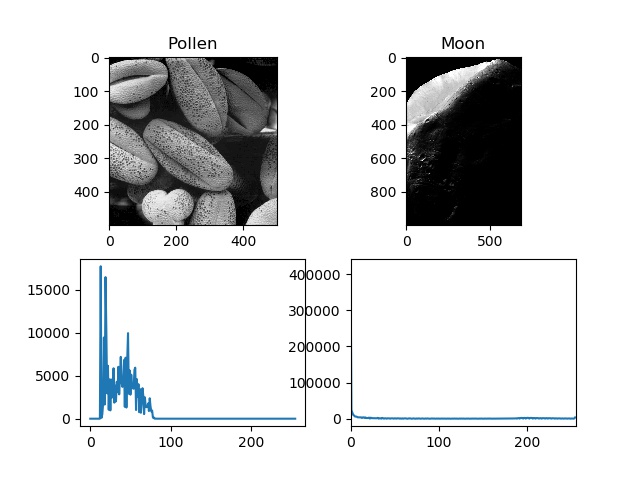
1. **Original and Equalized Image of Pollen**

**** 

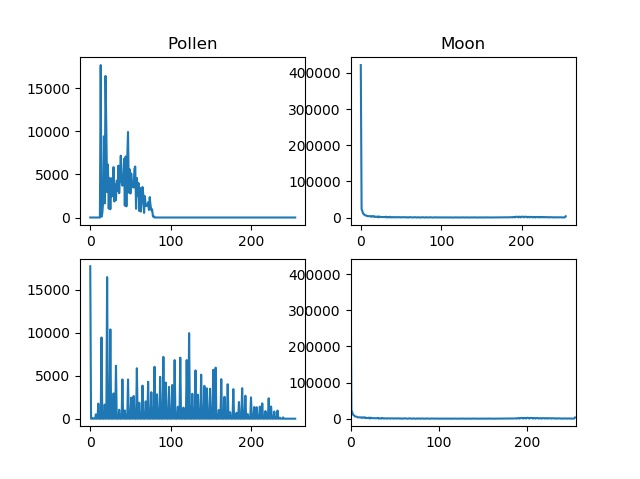
1. **Original and Equalized Image of Moon**

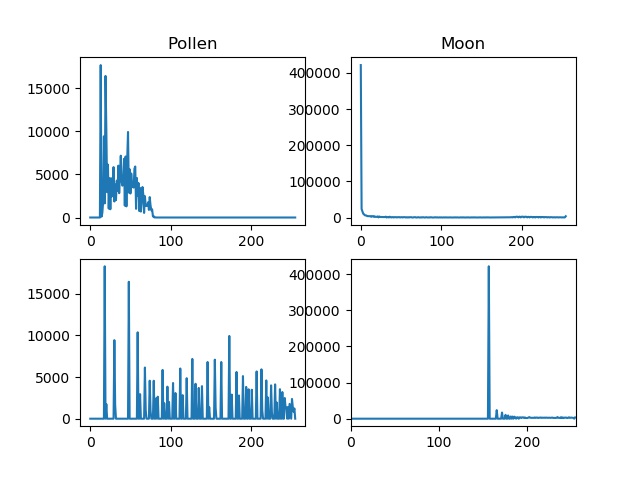
1. **Histogram of the Originals and It’s Images**



1. **Histogram of Original and Scaled Images**



1. **Histogram of Original and Equalized Images**

****

**Conclusion**

1. **Matlab**

* Scale 처리의 경우 **Pollen은 좋은 이미지**가 되었고, 이를 히스토그램에서 확인할 수 있었다. 그러나 **Moon은 큰 차이가** 없었다. 그 이유는 히스토그램을 보면 기존의 Moon 이미지 픽셀 값 자체가 **넓게 분포**되어 있기 때문이고, 특히 **어두운 값(1에 가까운 값) 밝은 값(255에 가까운 값)에 집중 분포** 되어있기에 Scale 처리 효과가 미미하다.
* **Equalization은 이미지 전체 밝기를 전반적으로 밝게** 해준다. 따라서 Moon의 경우에도 Scale에서 미처 하지 못했던 기능을 수행한다. 다만 전반적 밝기가 밝아져, Contrast가 약해졌다. 그럼에도 불구하고 **어두운 부분의 이미지 판별**이 필요한 경우가 있어 **용이한 사진**이다.
* **Main Code에서 Scale 범위 찾을 때 find 함수 쓴 이유:** 단순히 히스토그램에서 0이 아닌 값의 시작과 끝을 찾는다는 의미를 주기 위해서이다. 그러나 굳이 find 쓸 필요 없이 이미지에 **min(min())** 또는 **max(max())**를 쓰면 컴퓨터적으로 계산이 더 쉽다.
* Equalization할 때에는 Scale과 다르게 함수 식을 세워 대입하는 것이 아니기에 **CDF**에 **이미지 값**을 직접 **for구문을 써서 대입**해준다.

1. **Python**

* 이번 python의 경우 library로 cv2, numpy, matplotlib를 썼다. 특히 **matplotlib**의 경우 매트랩과 비슷한 개발 환경을 제공해주는 라이브러리이기에 **두 영상 간의 변화를 비교하기에 좋다**.
* Matplotlib를 이용했기에 **plt** 함수를 이용하여, Matlab처럼 **figure, subplot을 만드는 것이 가능**해짐.
* **Matplotlib**은 주로 **데이터를 시각화** 할 때 많이 쓰인다.