초급 영상처리

(나만의 Opencv 구현하기)

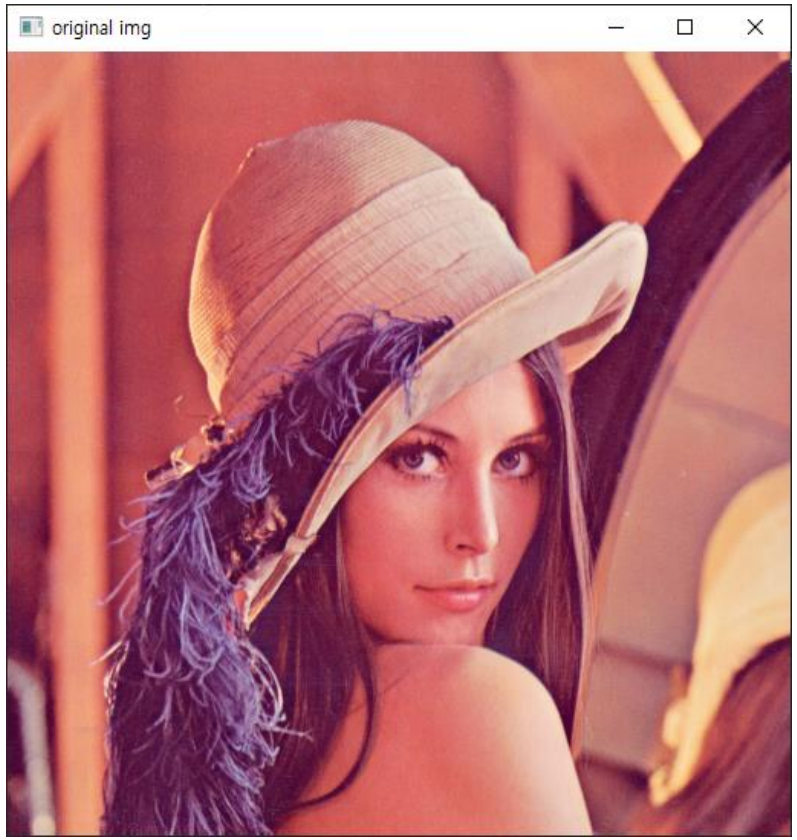
박화종

INDEX

- 저번 주 과제 정답
- Histogram
- 실습
- 과제

저번 주 과제(IP1_test1)

- 컬러 영상 변환하기



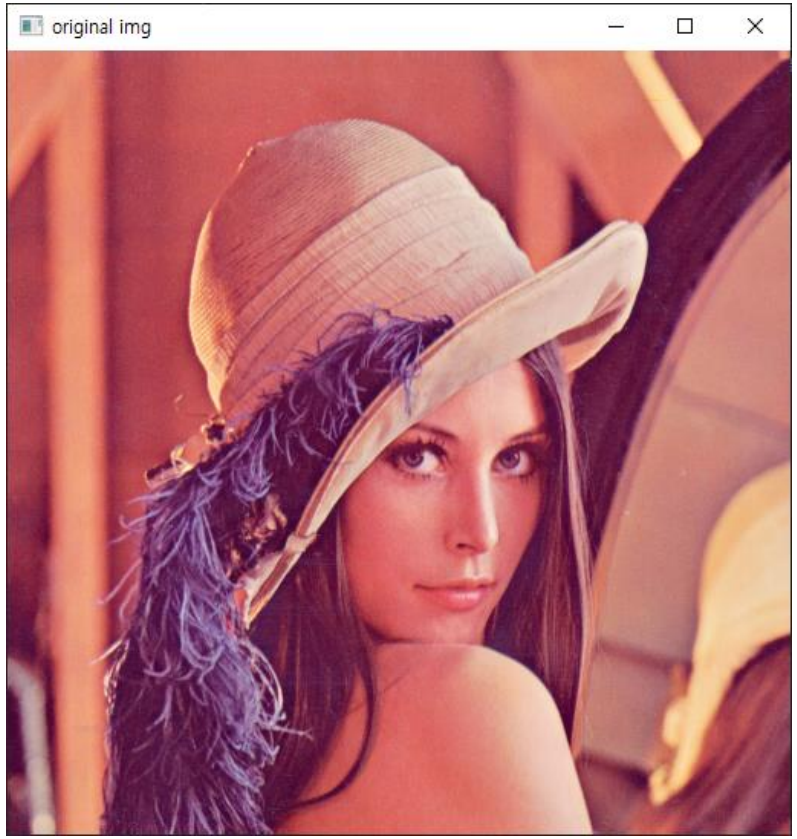
저번 주 과제(IP1_test1)

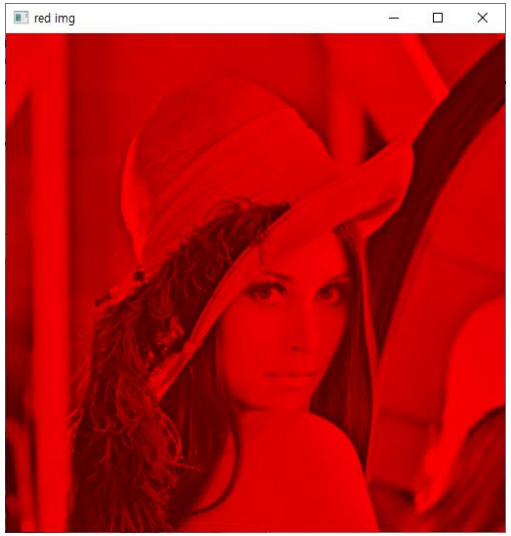
- 컬러 영상 변환하기
  - 과연 B, G, R 각각 1/3씩 사용하는 것이 맞는가?

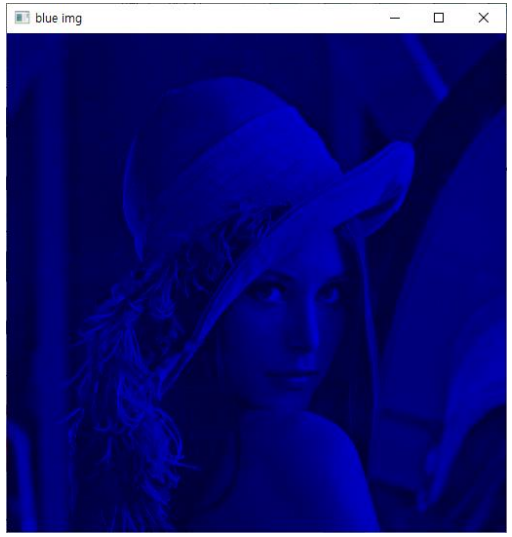
```python
def main():
    img = cv2.imread('lena.png')
    cv2.imshow('original img', img)

    gray = BGR2Gray(img)
    cv2.imshow('gray img', gray.astype(np.uint8))

    cv2.waitKey()
    cv2.destroyAllWindows()

def BGR2Gray(img):
    img = img.astype(np.float32)
    gray = img[..., 0]*0.1140 + img[..., 1]*0.5870 + img[..., 2]*0.2989
    return gray
```

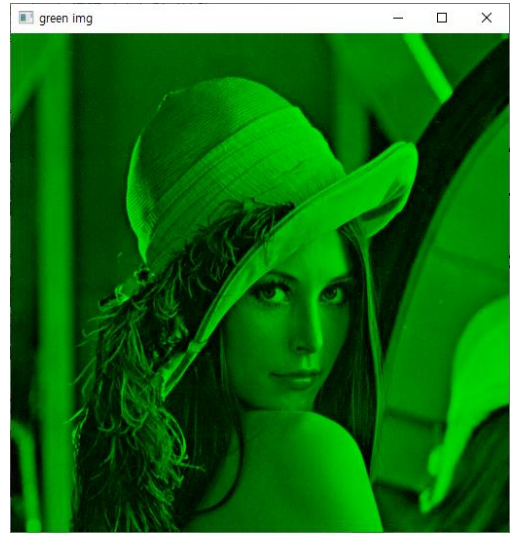
저번 주 과제(IP1_test2)

- 컬러 영상 변환하기2









저번 주 과제(IP1_test2)

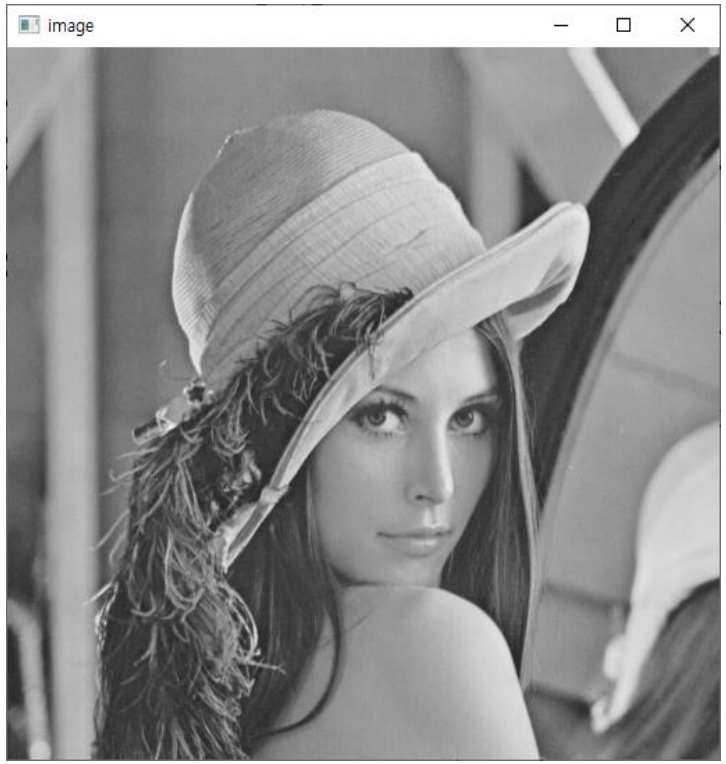
- 컬러 영상 변환하기2
  - .copy() 붙인 이유도 찾아보기

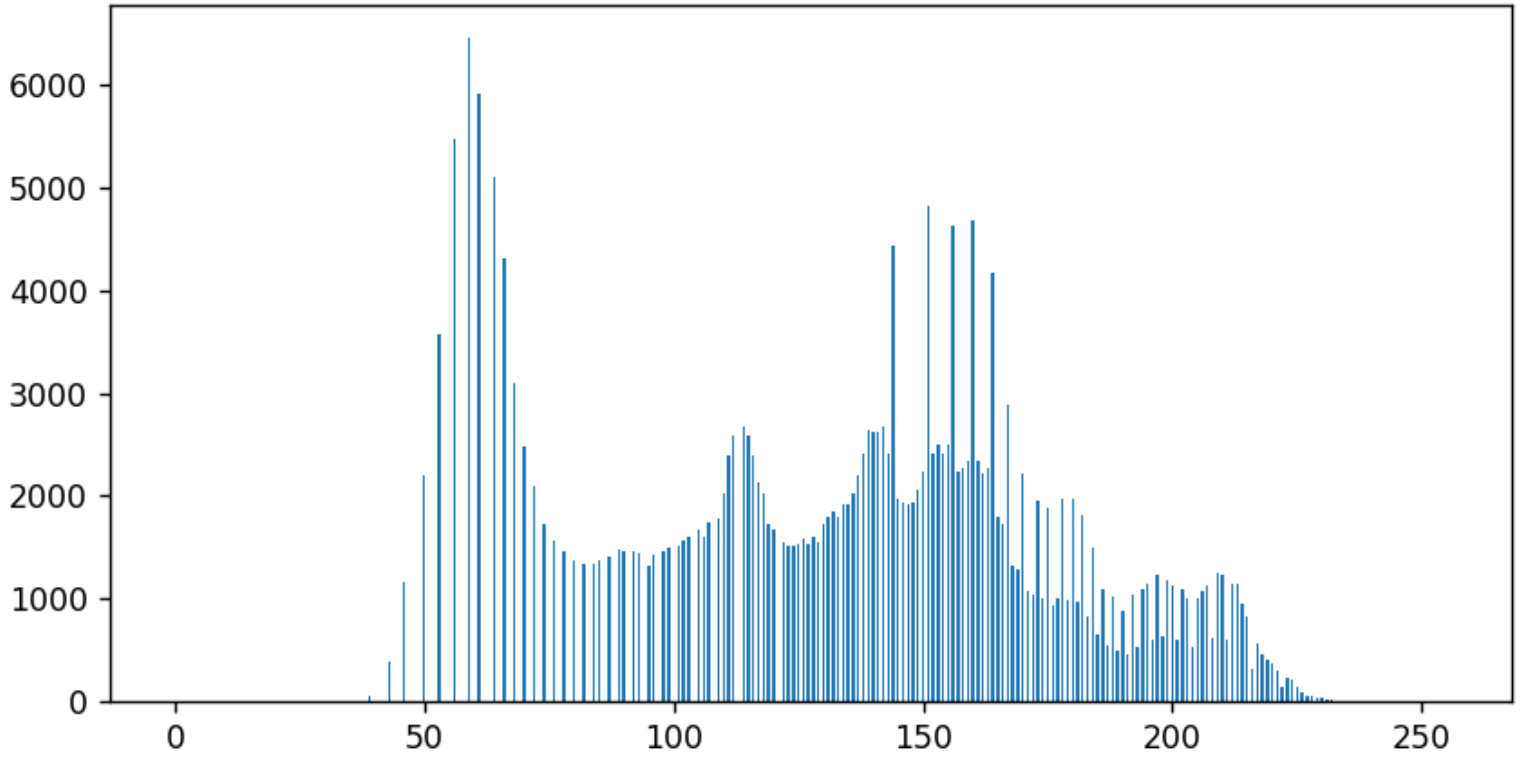
```python
def main():
    img = cv2.imread('lena.png')
    cv2.imshow('original img', img)

    red = BGR2Red(img.copy())
    cv2.imshow('red img', red)

    green = BGR2Green(img.copy())
    cv2.imshow('green img', green)

    blue = BGR2Blue(img.copy())
    cv2.imshow('blue img', blue)

    cv2.waitKey()
    cv2.destroyAllWindows()
```

```python
def BGR2Red(img):
    img[..., (0,1)] = 0
    return img

def BGR2Green(img):
    img[..., (0,2)] = 0
    return img

def BGR2Blue(img):
    img[..., (1,2)] = 0
    return img
```

Histogram

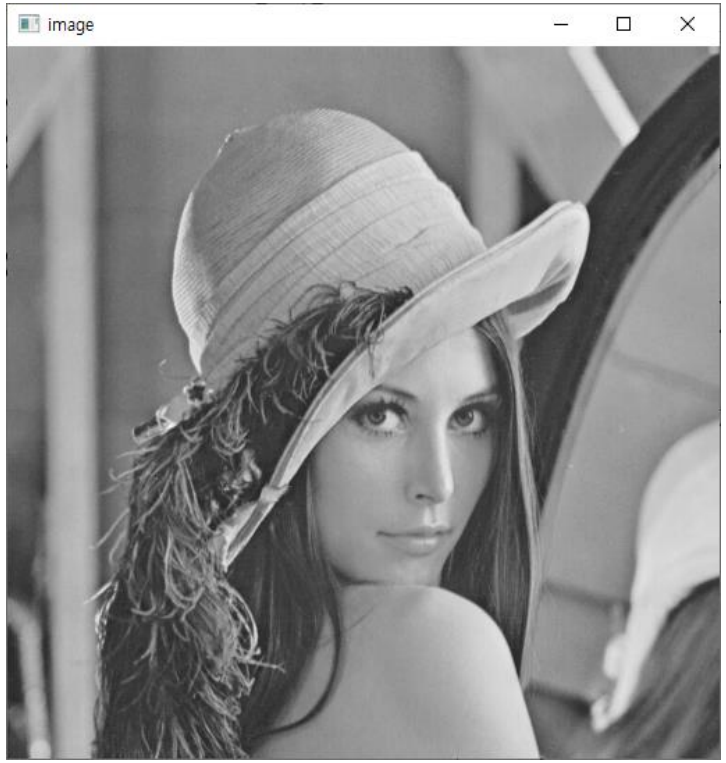
- Histogram이란?
  - Intensity에 대한 빈도 수를 그래프로 나타낸 것





Histogram

- Histogram 그리기
  - 이미지 변환하기



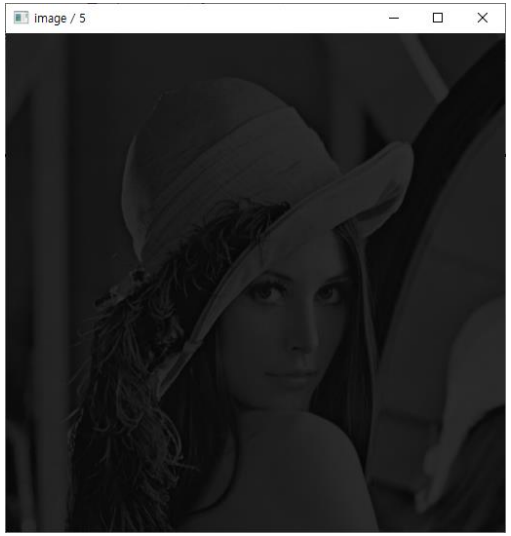


Image / 5

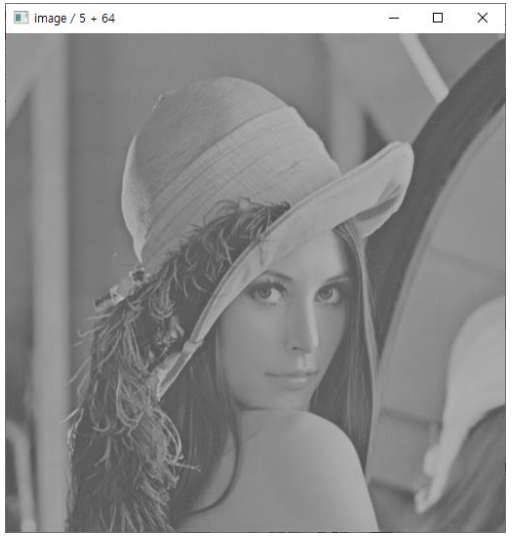


Image / 5 + 64

Image * 2

Histogram

- Histogram 그리기

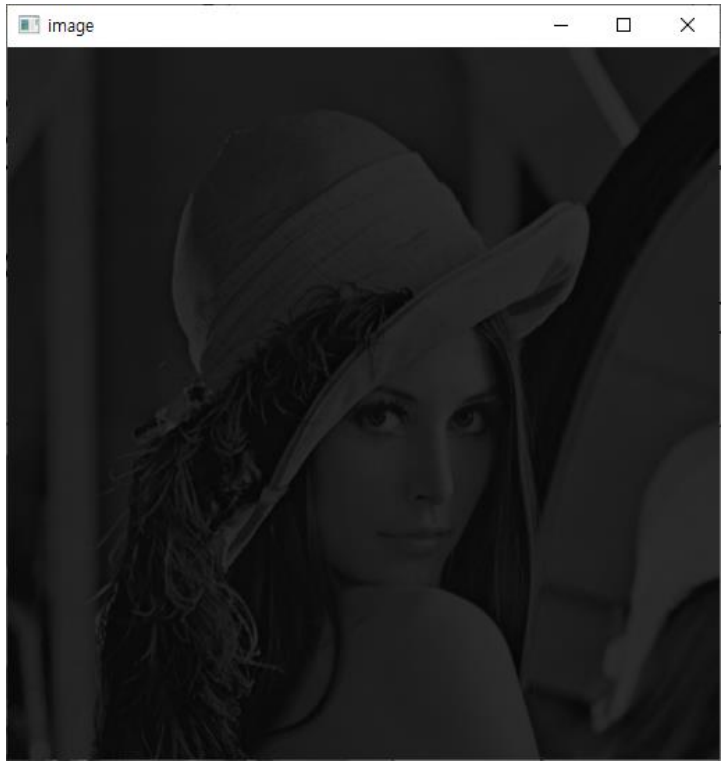
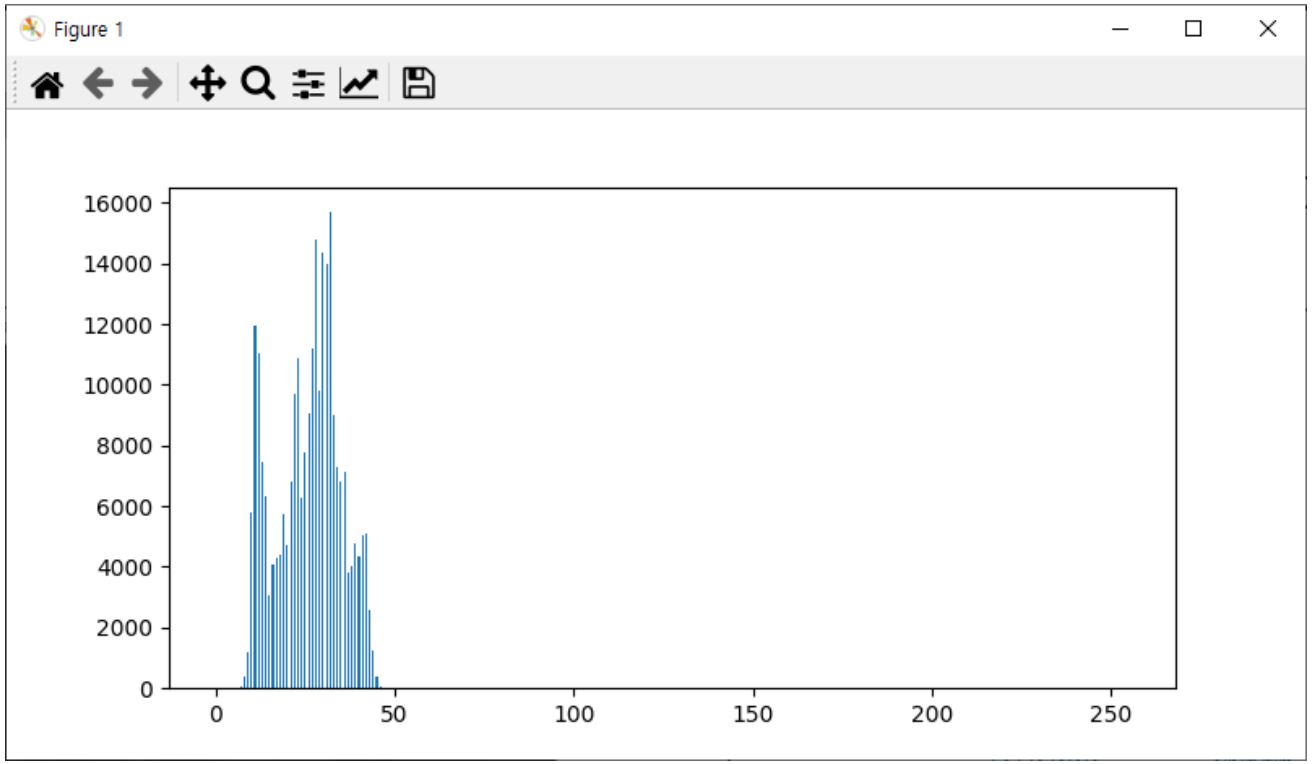


Image / 5



Histogram

- Histogram 그리기

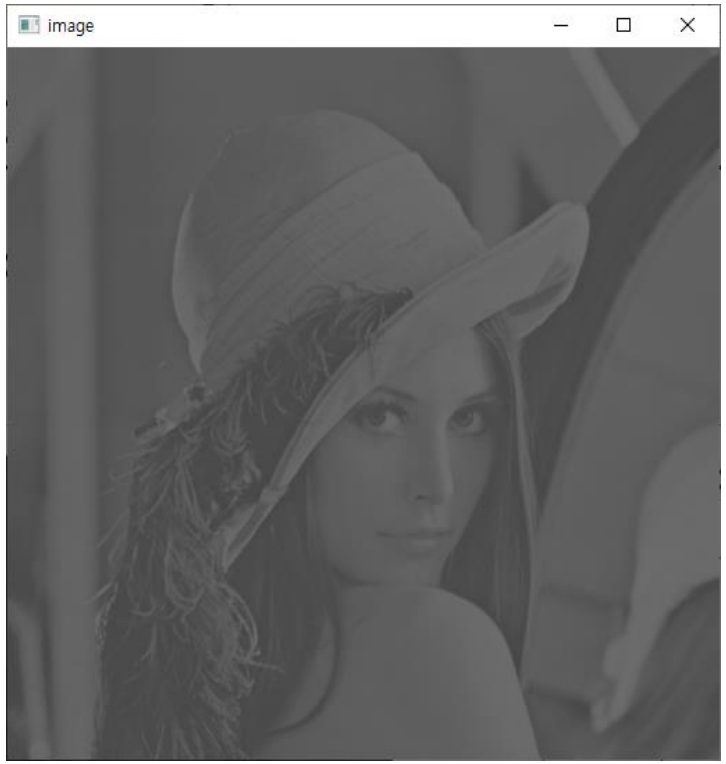
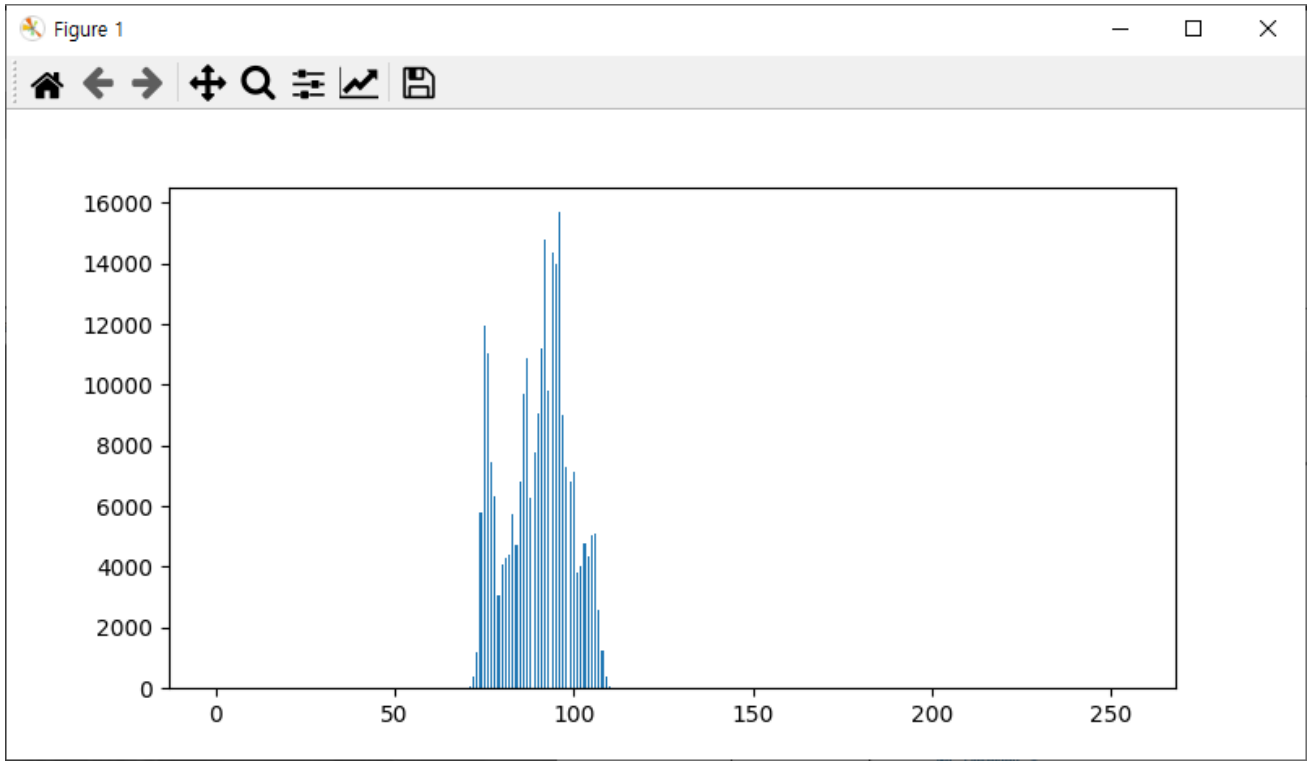


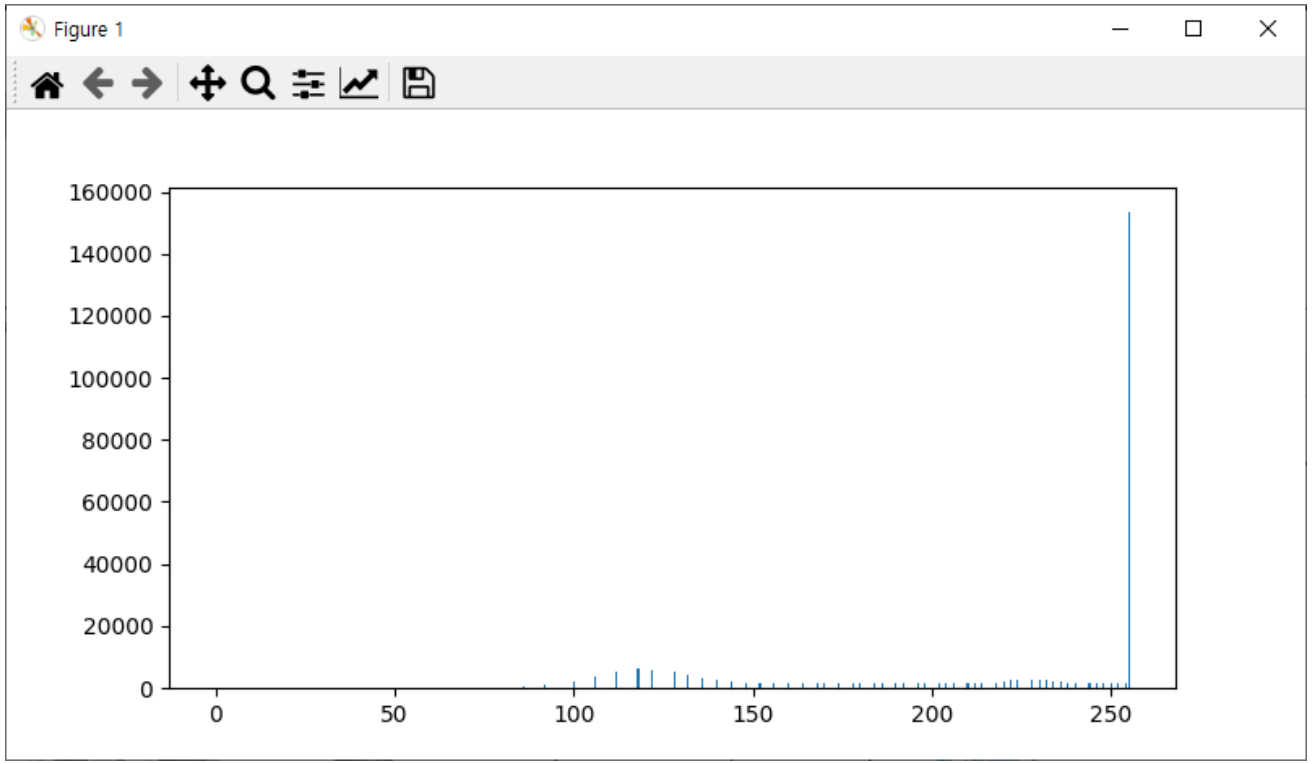
Image / 5 + 64



Histogram

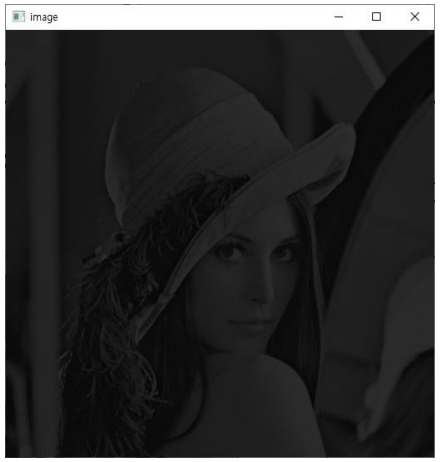
- Histogram 그리기

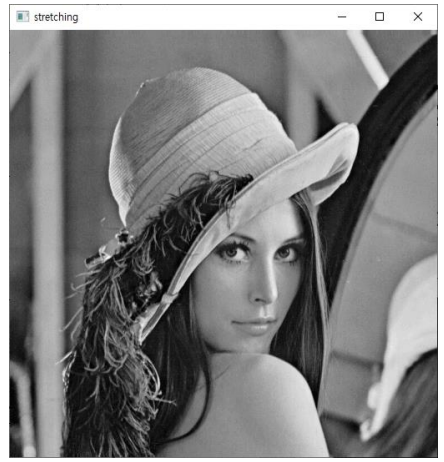
Image * 2

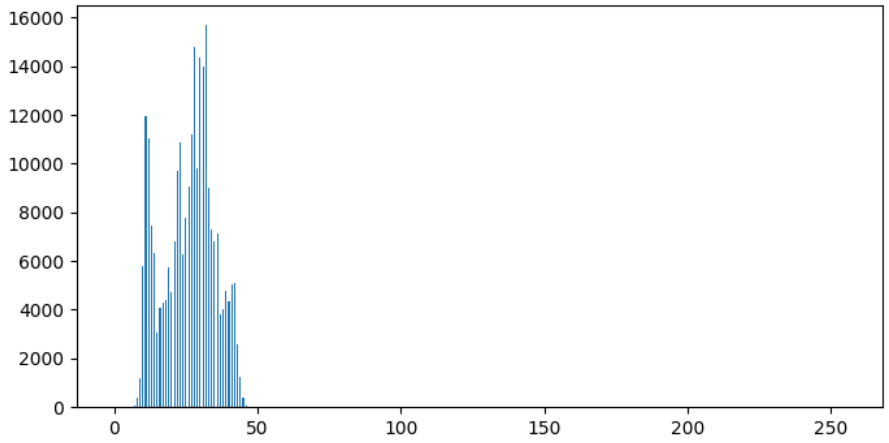


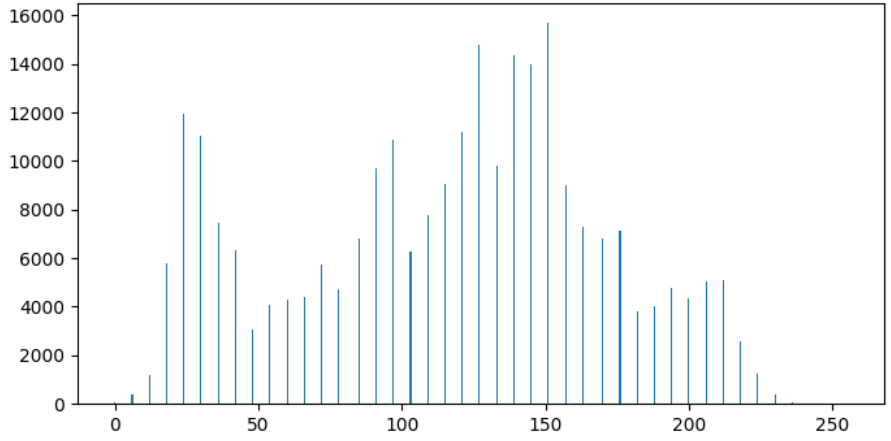
Histogram

- Histogram stretching
  - Histogram을 균일하게 만드는 작업



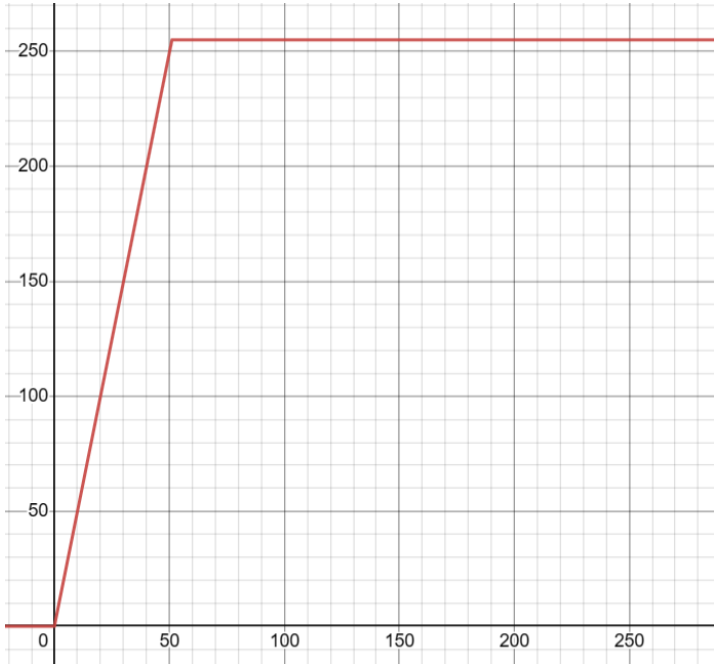




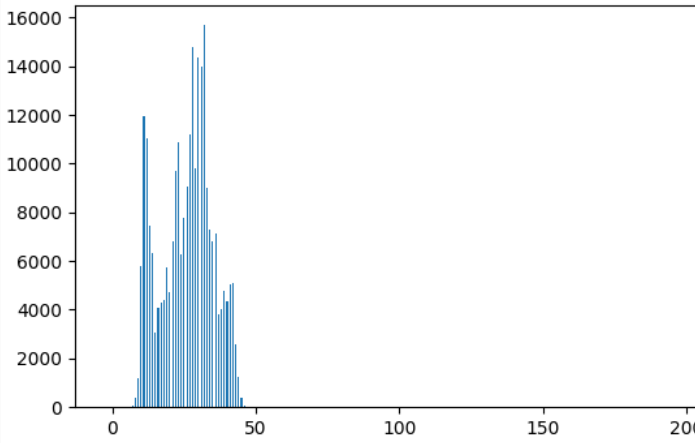


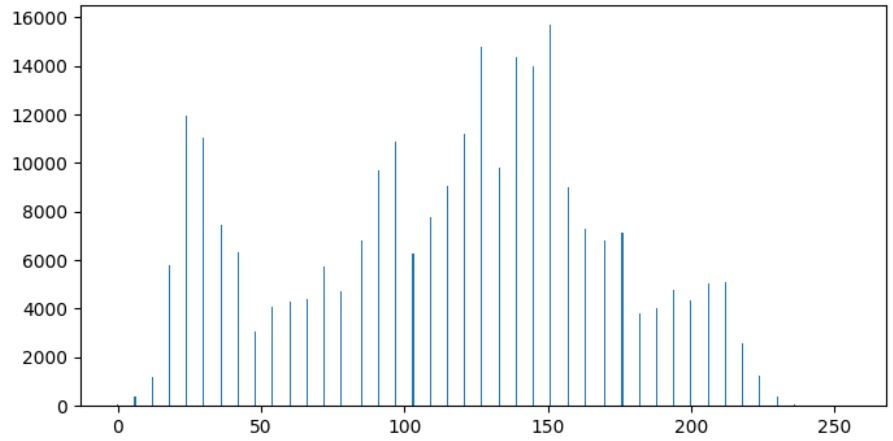
Histogram

- Histogram stretching
  - Histogram을 균일하게 만드는 작업
  - $x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \times 255$



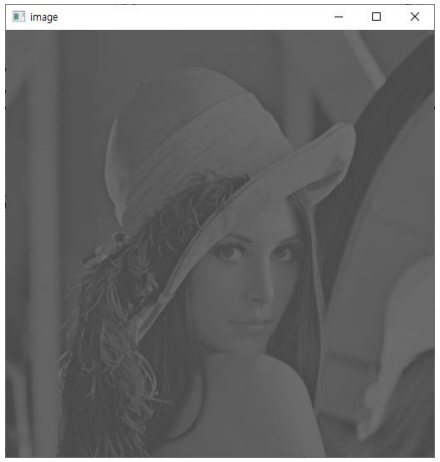
Stretching function

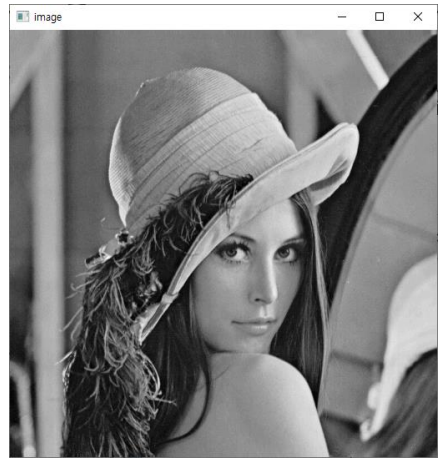


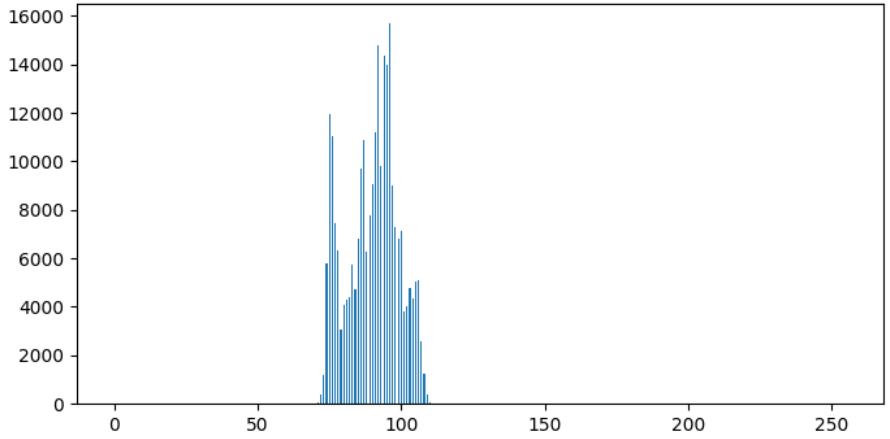


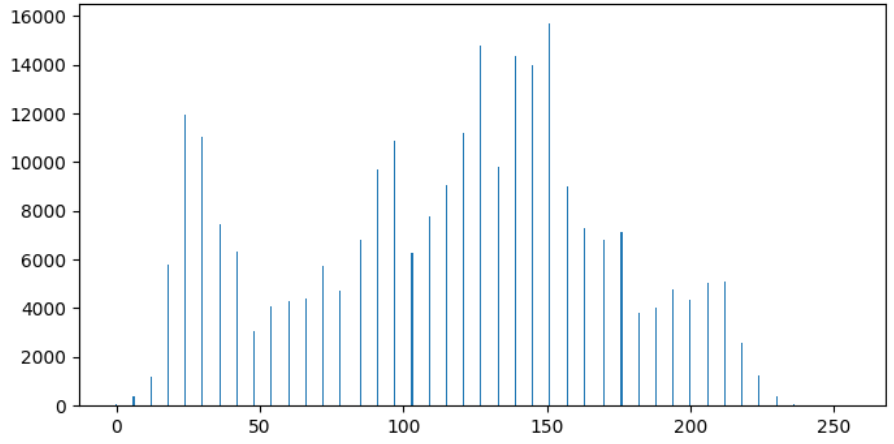
Histogram

- Histogram stretching
  - Histogram을 균일하게 만드는 작업



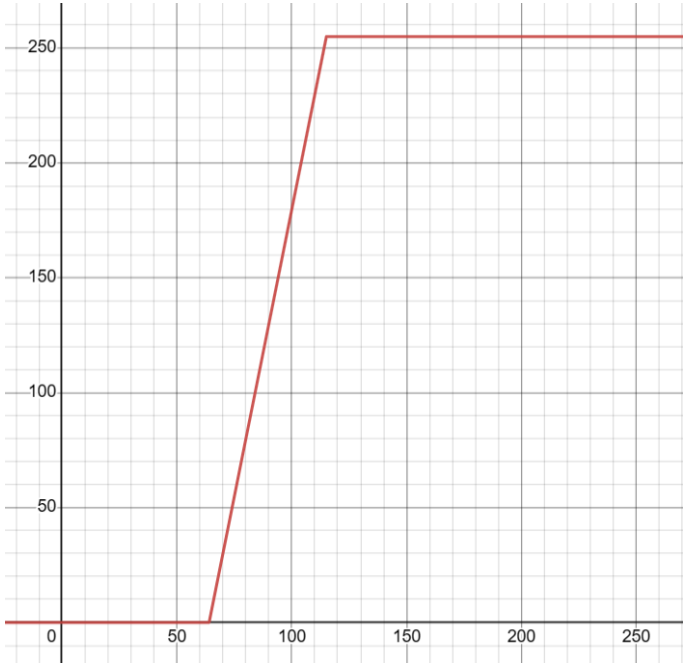




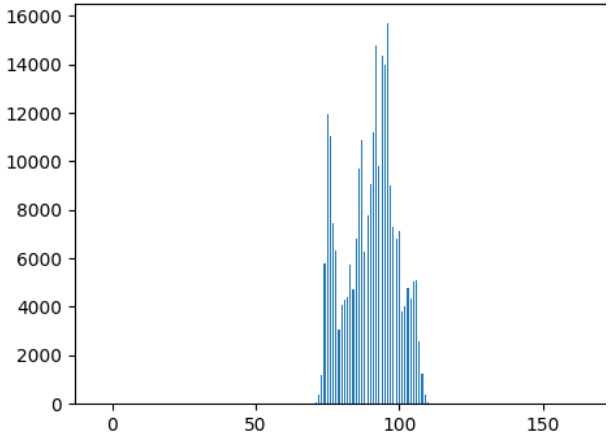


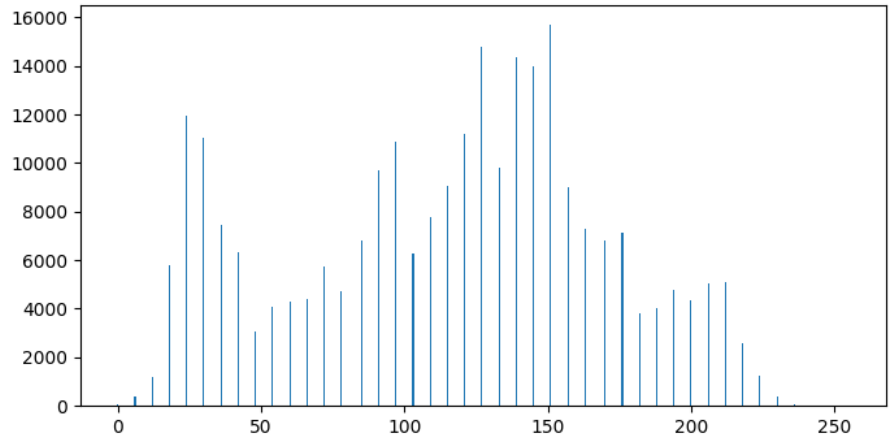
Histogram

- Histogram stretching
  - Histogram을 균일하게 만드는 작업
  - $x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \times 255$



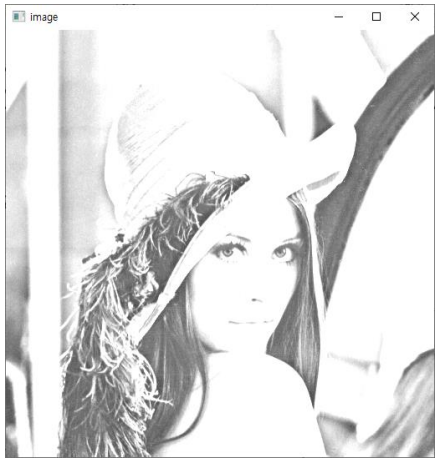
Stretching function

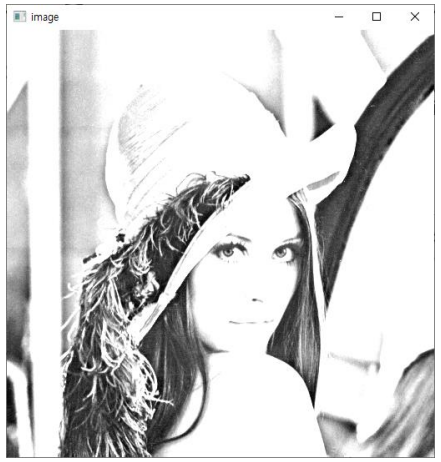


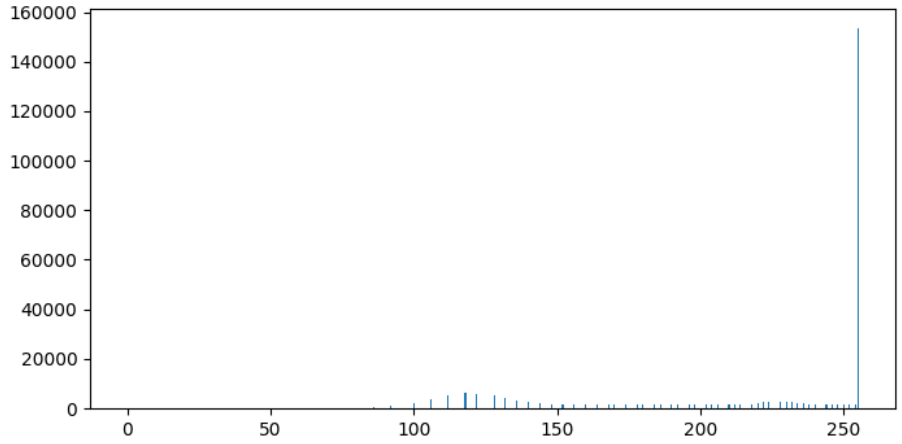


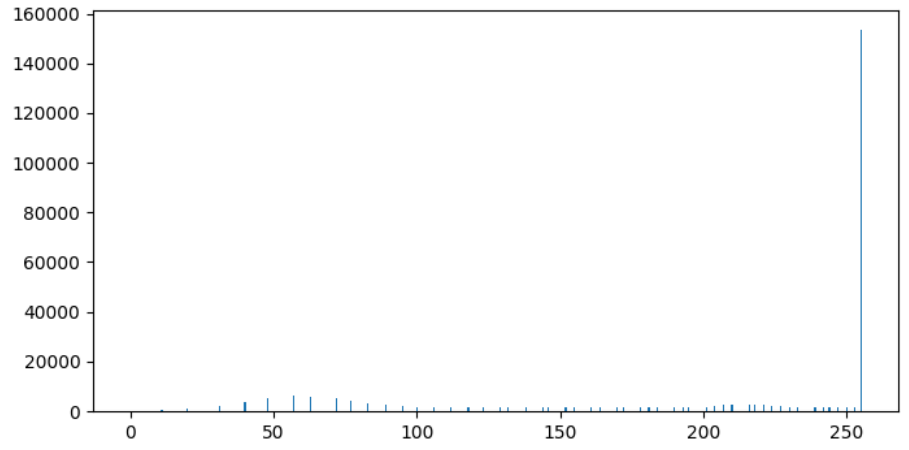
Histogram

- Histogram stretching
  - Histogram을 균일하게 만드는 작업



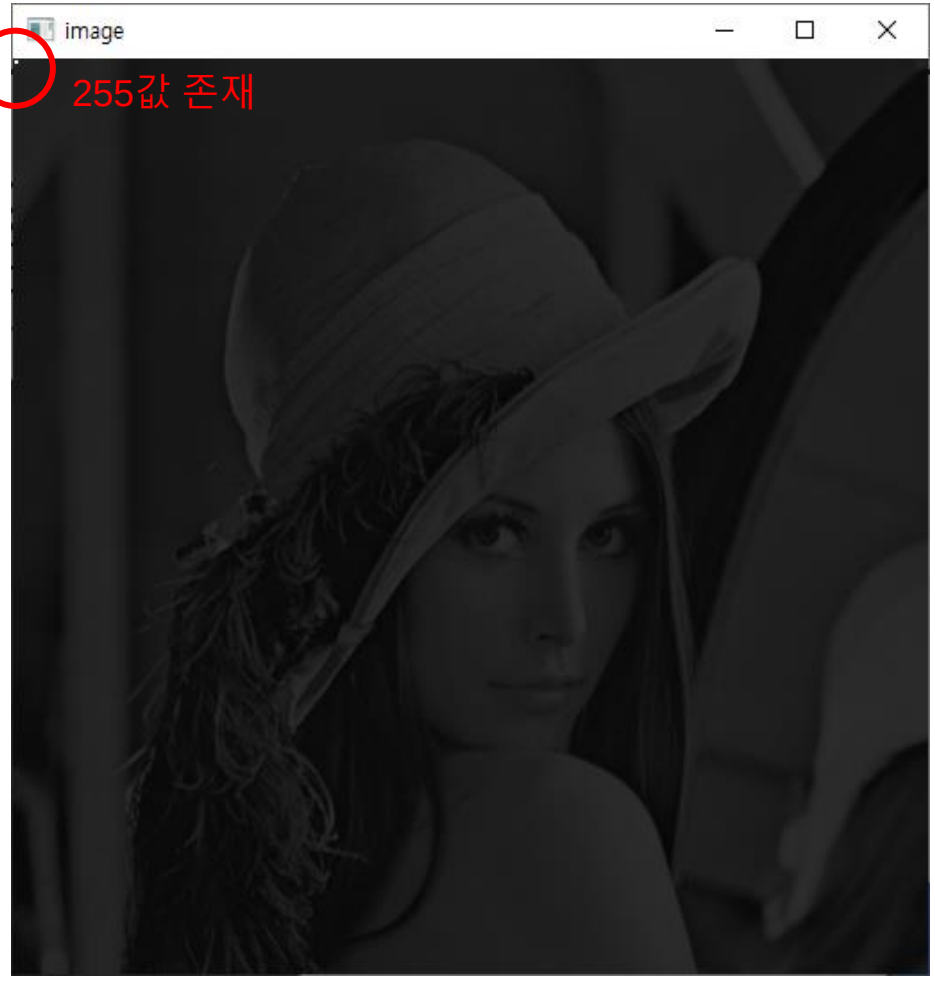






Histogram

- Histogram stretching
  - Stretching을 진행하면 어떻게 될까?



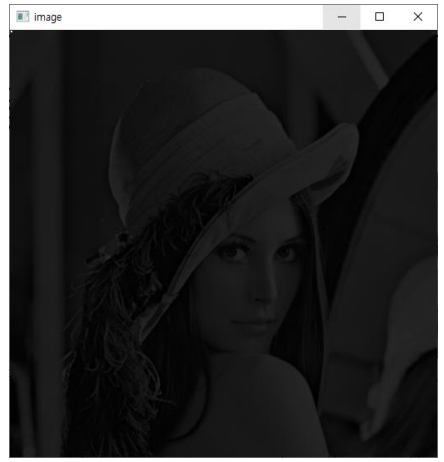
```
img = img / 5
img[1:3,1:3] = 255
img = np.clip(img, 0, 255)
img = img.astype(np.uint8)
```

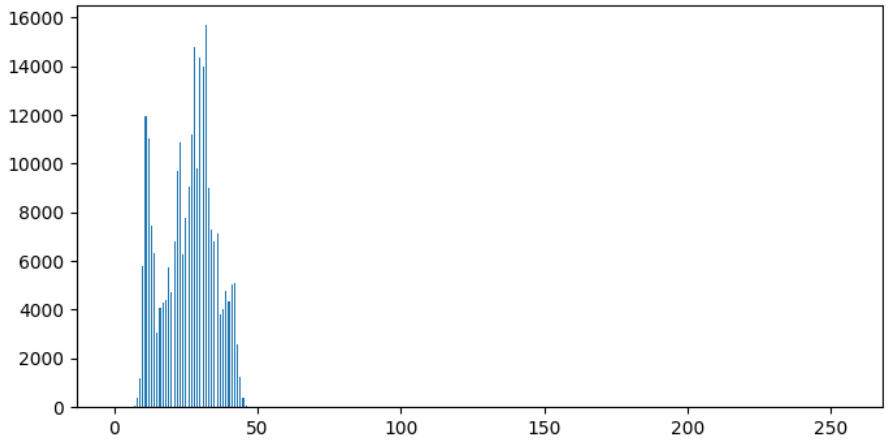
Histogram

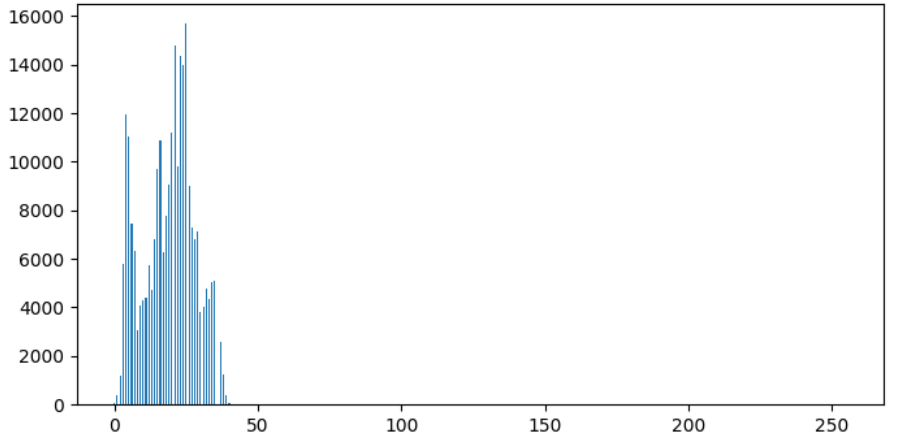
- Histogram stretching
  - 원하는 결과가 나오지 않음

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \times 255$$

Stretching



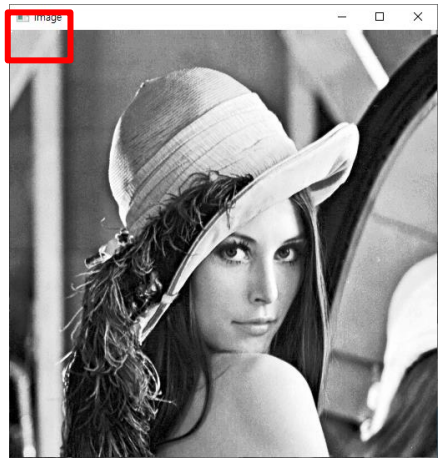


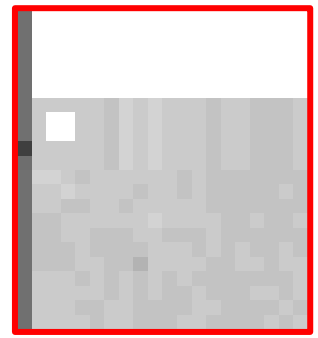


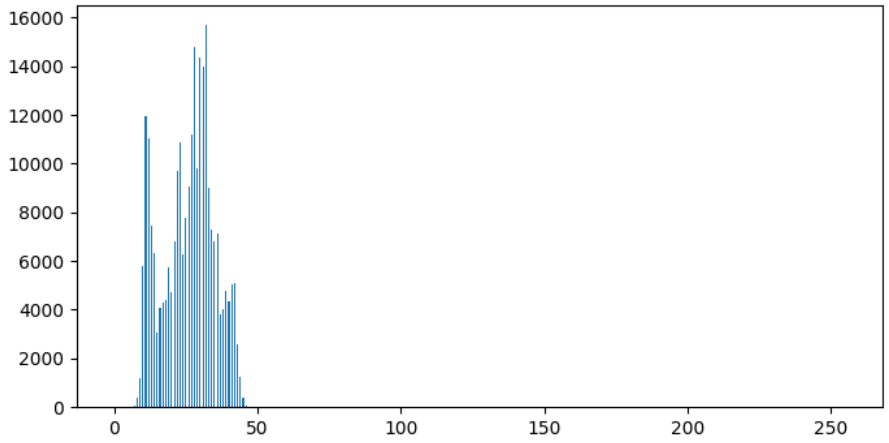
Histogram

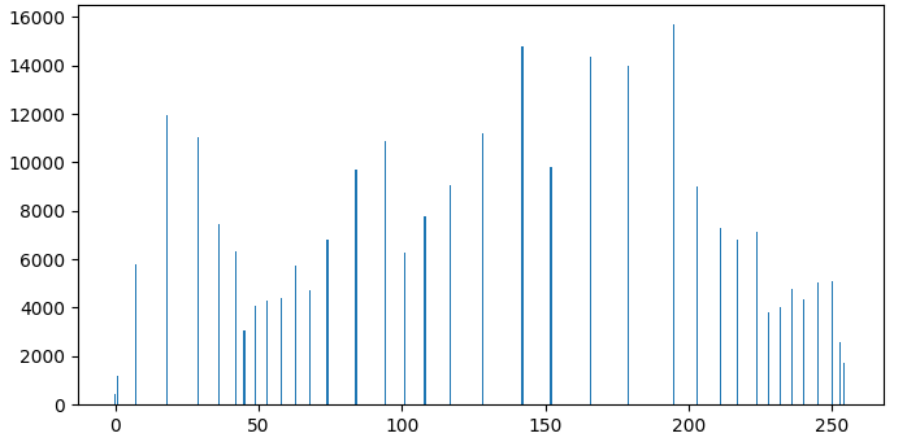
- Histogram equalization
  - Intensity의 누적 분포 함수를 사용하여 영상을 개선

Equalization



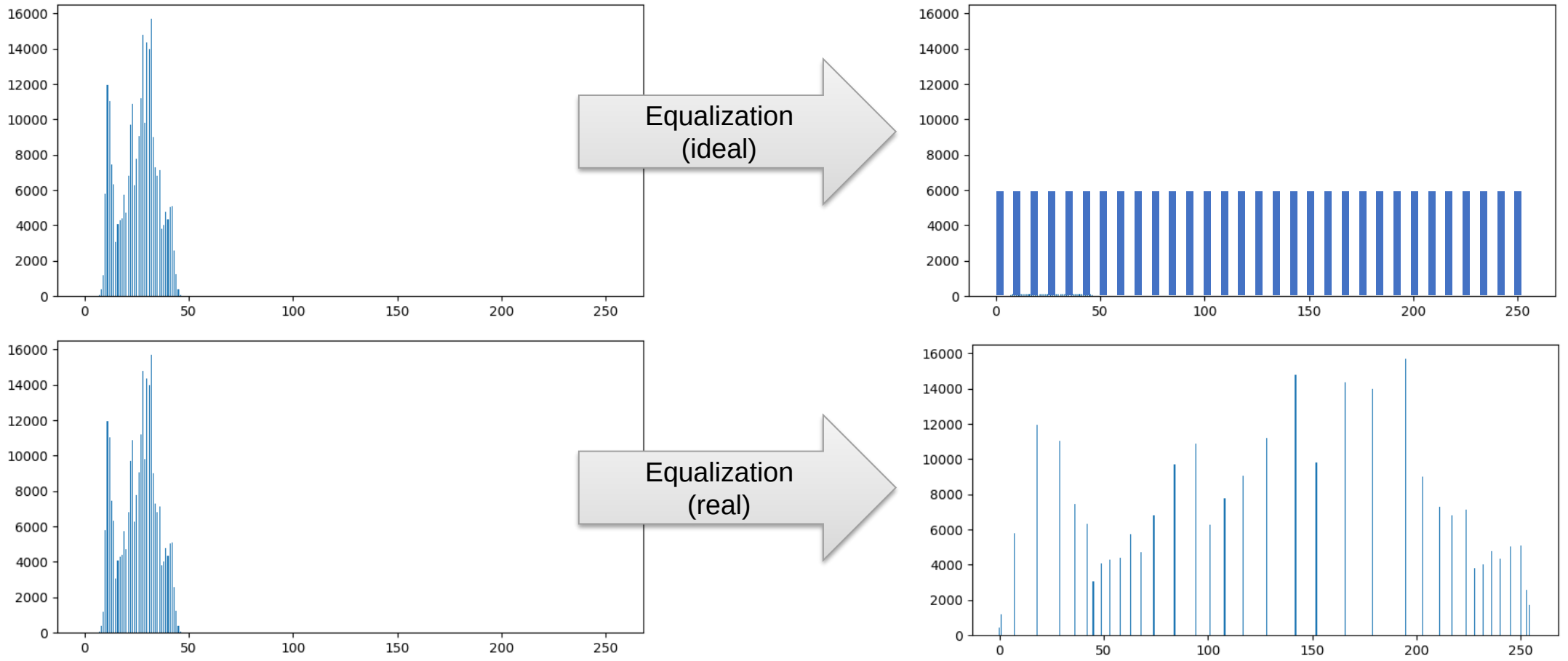






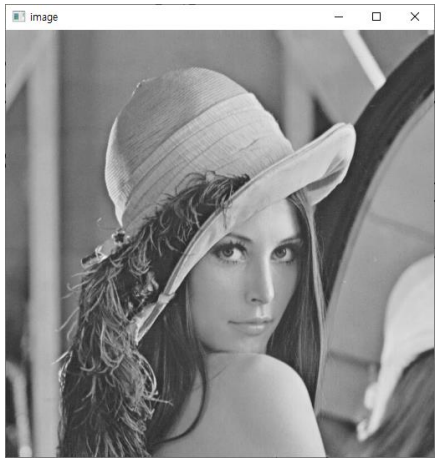
Histogram

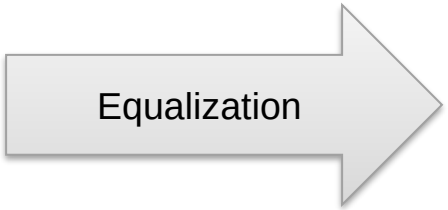
- Histogram equalization
  - Intensity의 누적 분포 함수를 사용하여 영상을 개선

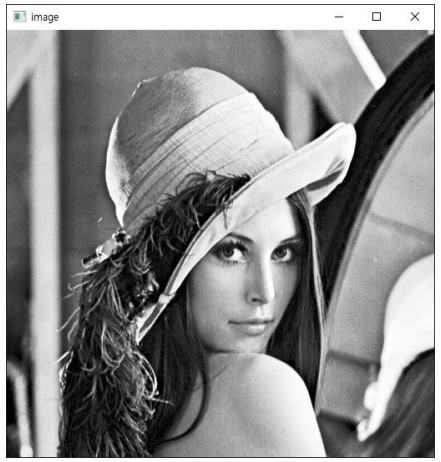


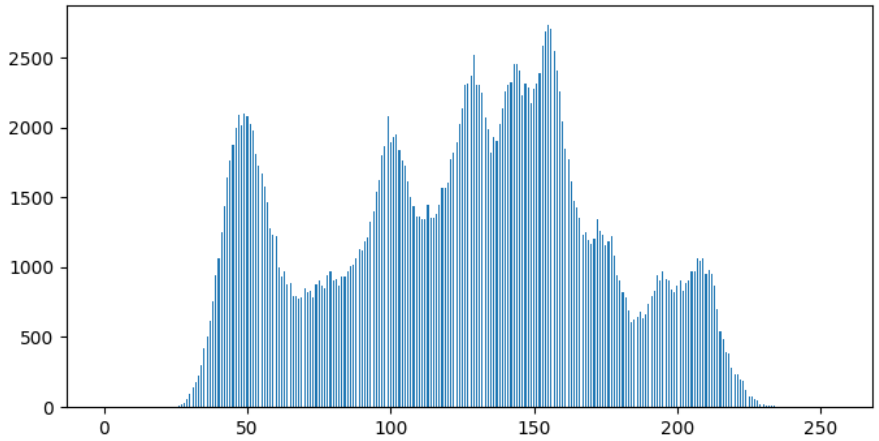
Histogram

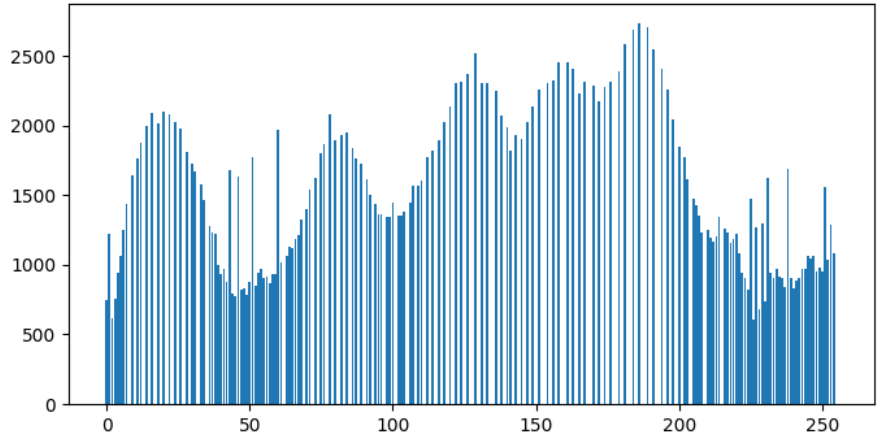
- Histogram equalization
  - Intensity의 누적 분포 함수를 사용하여 영상을 개선









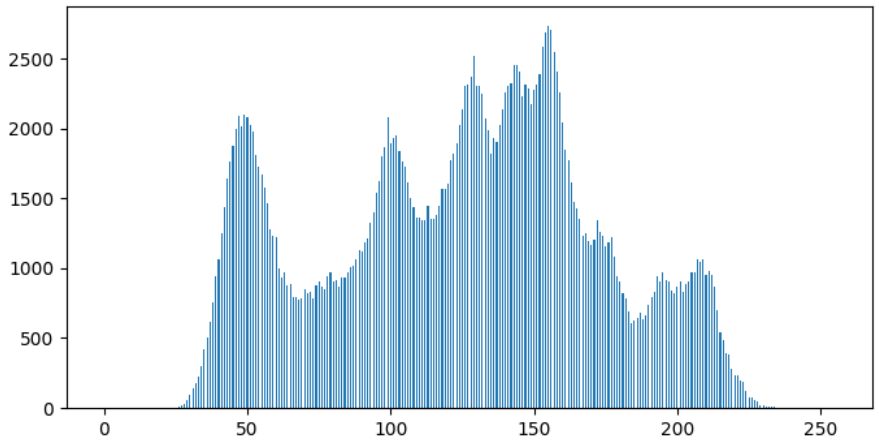


Histogram

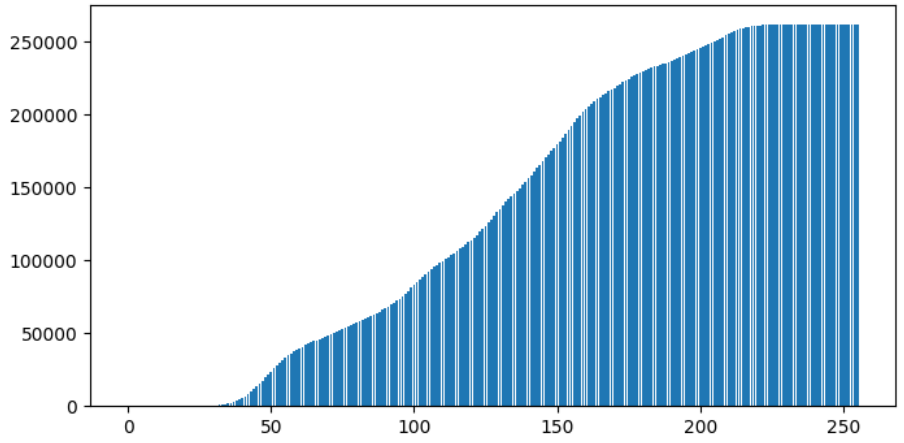
- Histogram equalization 계산 과정
  - 누적 히스토그램 구하기
    - 히스토그램 구하기
    - 구해진 히스토그램으로 누적 히스토그램으로 변환
  - 정규화 하기(0 ~ max intensity)
    - 0 ~ 1로 정규화하기
    - Max intensity값 곱하기
    - Float -> int (반올림, 내림 등 어떤 것을 사용하던 크게 상관 없음)
  - 정규화 된 데이터를 사용하여 히스토그램 다시 계산하기
    - 다시 히스토그램 구하기

Histogram

- Histogram equalization 계산 과정
  - 누적 히스토그램 구하기
    - 히스토그램 구하기
    - 구해진 히스토그램으로 누적 히스토그램으로 변환



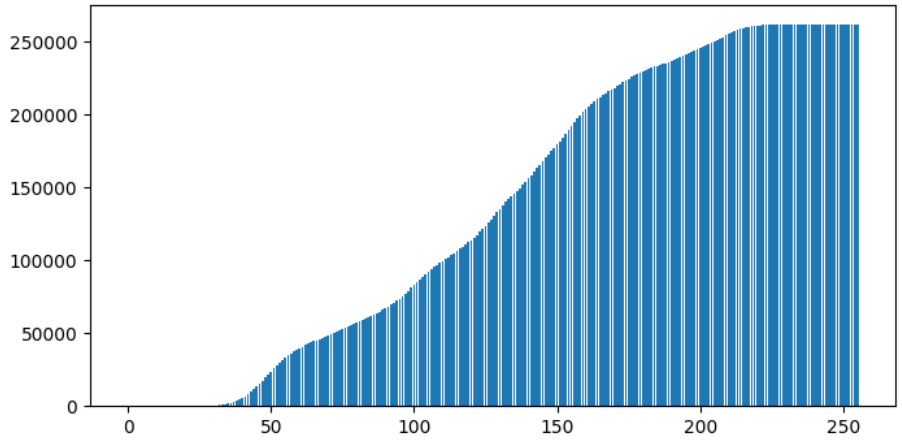
히스토그램



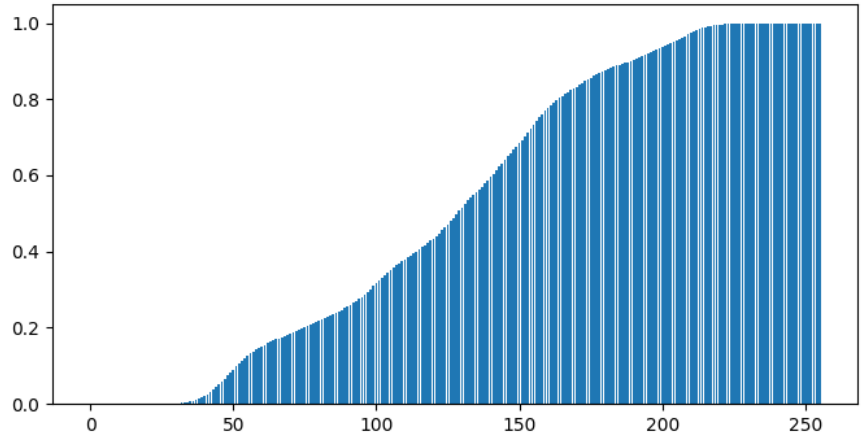
누적 히스토그램

Histogram

- Histogram equalization 계산 과정
  - 정규화 하기(0 ~ max intensity)
    - 0 ~ 1로 정규화하기
    - Max intensity값 곱하기
    - Float -> int (반올림, 내림 등 어떤 것을 사용하던 크게 상관 없음)



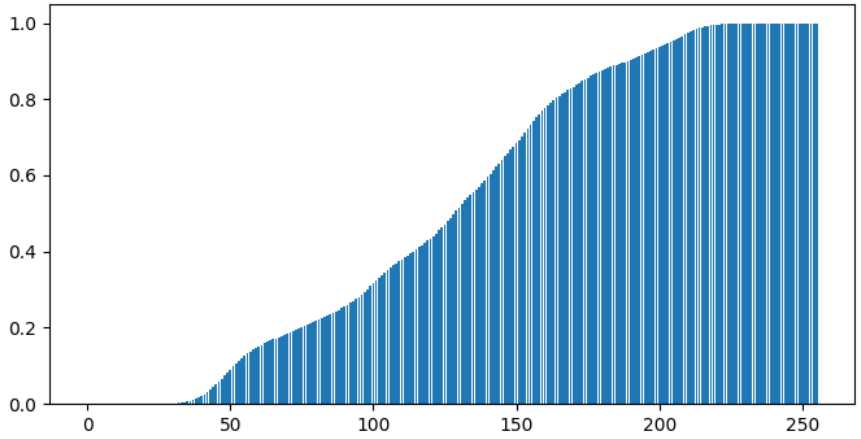
누적 히스토그램



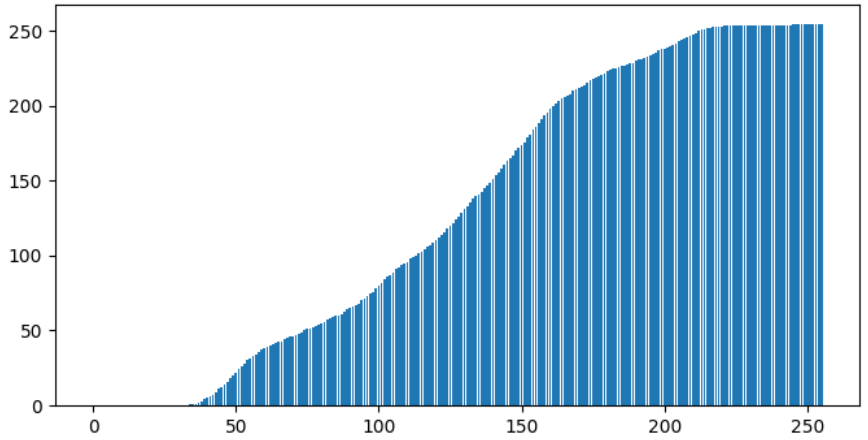
누적 히스토그램 정규화

Histogram

- Histogram equalization 계산 과정
  - 정규화 하기(0 ~ max intensity)
    - 0 ~ 1로 정규화하기
    - Max intensity값 곱하기
    - Float -> int (반올림, 내림 등 어떤 것을 사용하던 크게 상관 없음)



누적 히스토그램 정규화

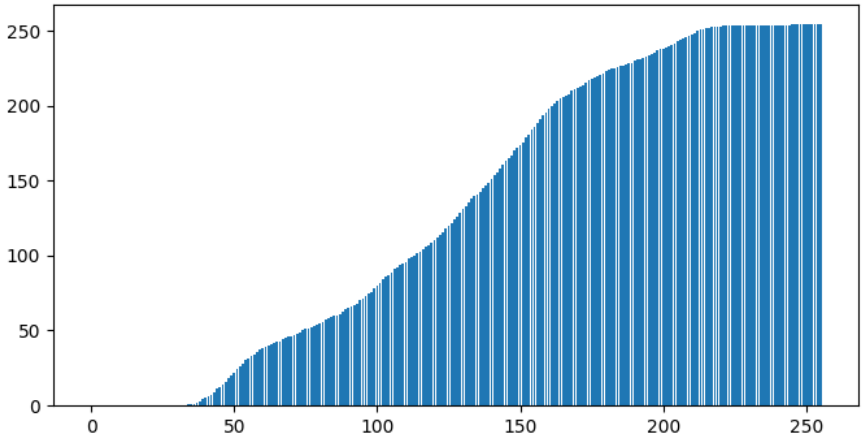


누적 히스토그램 정규화 * max intensity

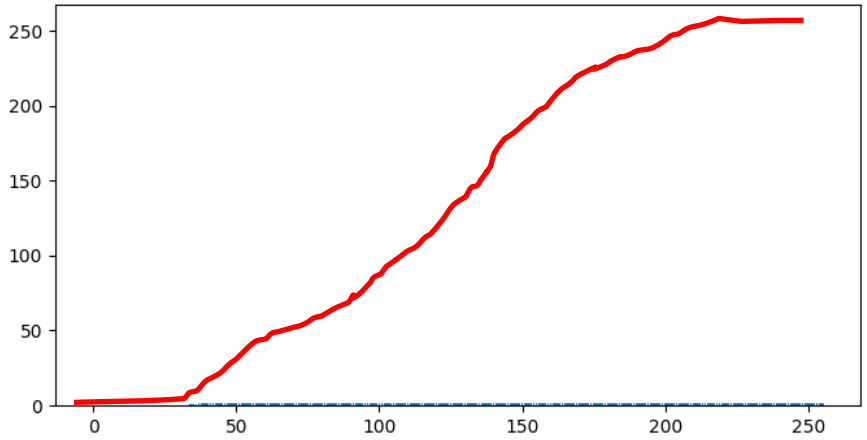
Histogram

- Histogram equalization 계산 과정
  - 정규화 된 데이터를 사용하여 히스토그램 다시 계산하기
    - 다시 히스토그램 구하기

이 그래프들의 의미는?



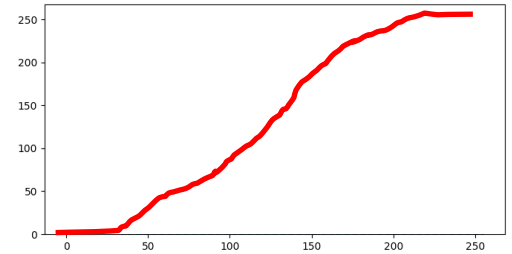
누적 히스토그램 정규화 * max intensity

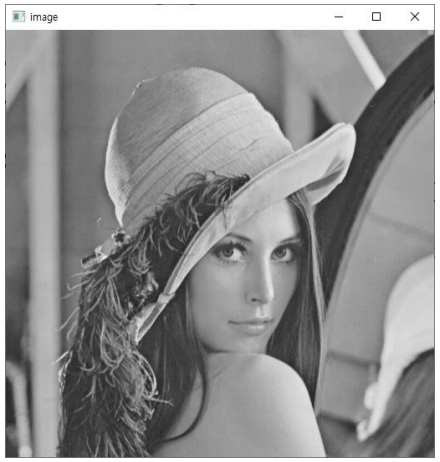


누적 히스토그램 정규화 * max intensity

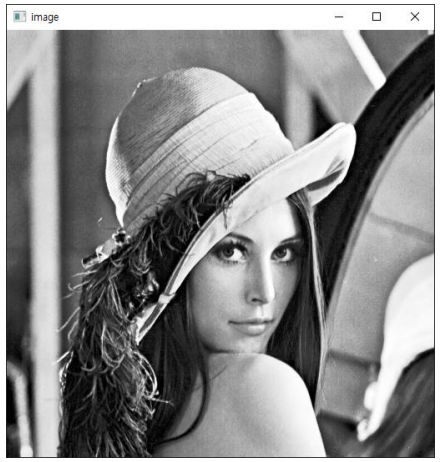
Histogram

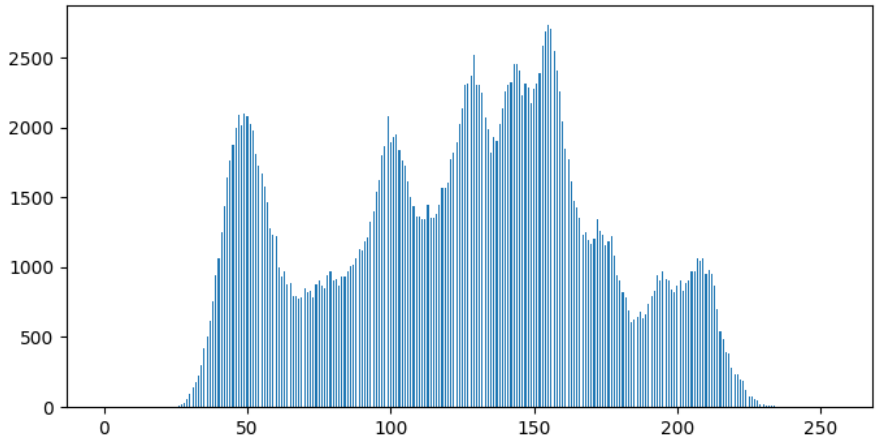
- Histogram equalization
  - Intensity의 누적 분포 함수를 사용하여 영상을 개선

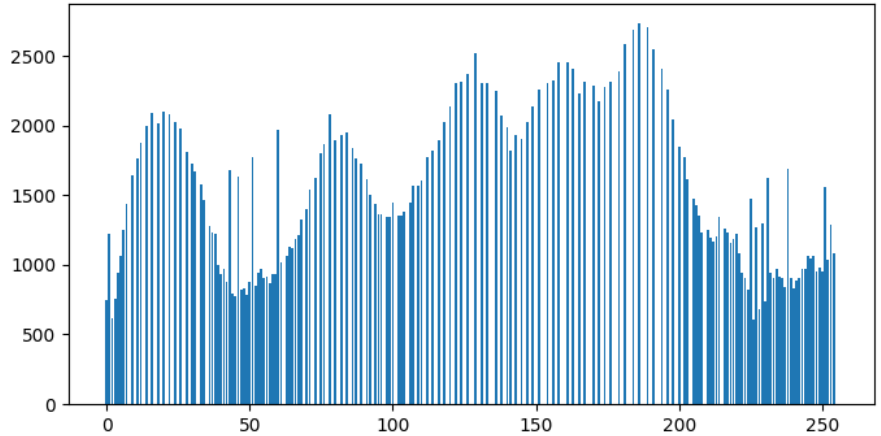




Equalization





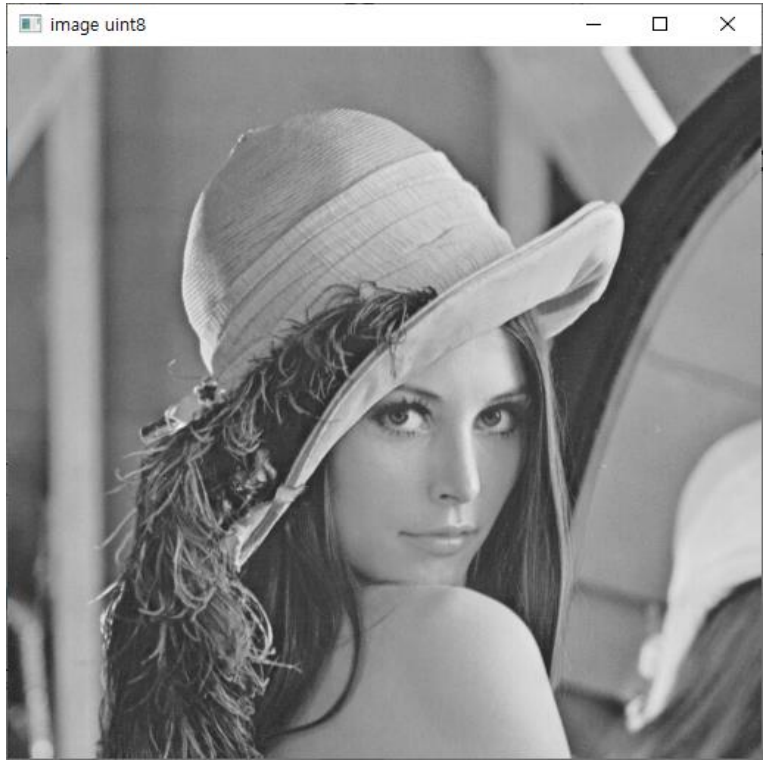


실습 및 과제

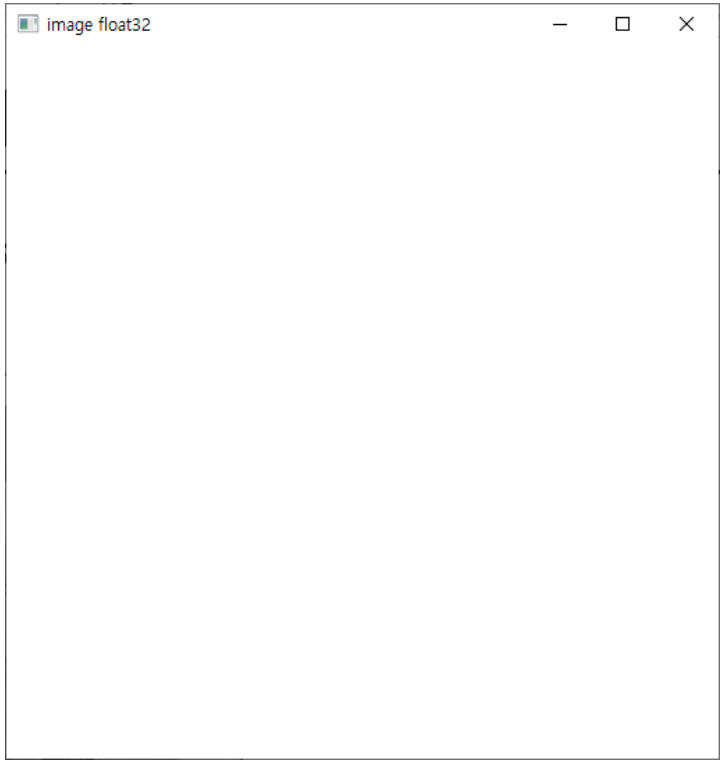
- Github : Hwa-Jong/MyOpenCV: study Opencv (github.com)

실습(IP2_1)

- cv2.imshow 함수에 대하여..



dtype = np.uint8



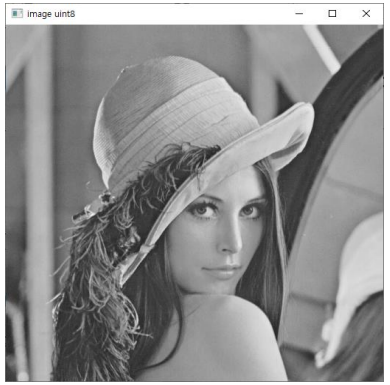
dtype = np.float32

dtype = np. float32

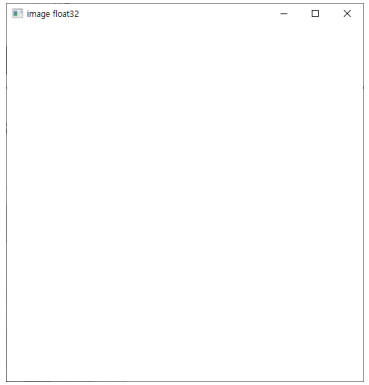
실습(IP2_1)

- cv2.imshow 함수에 대하여..

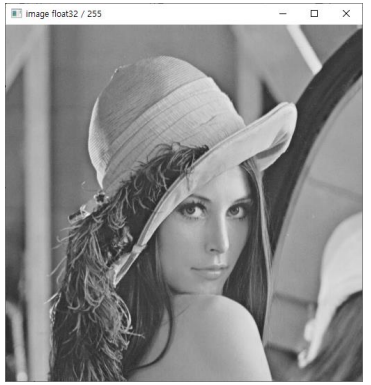
```python
def main():
    img_uint8 = cv2.imread('lena.png', cv2.IMREAD_GRAYSCALE)
    img_float32 = cv2.imread('lena.png', cv2.IMREAD_GRAYSCALE).astype(np.float32)

    cv2.imshow('image uint8', img_uint8)
    cv2.imshow('image float32', img_float32)
    cv2.imshow('image float32 / 255', img_float32/255)
    cv2.waitKey()
    cv2.destroyAllWindows()
```



dtype = np.uint8



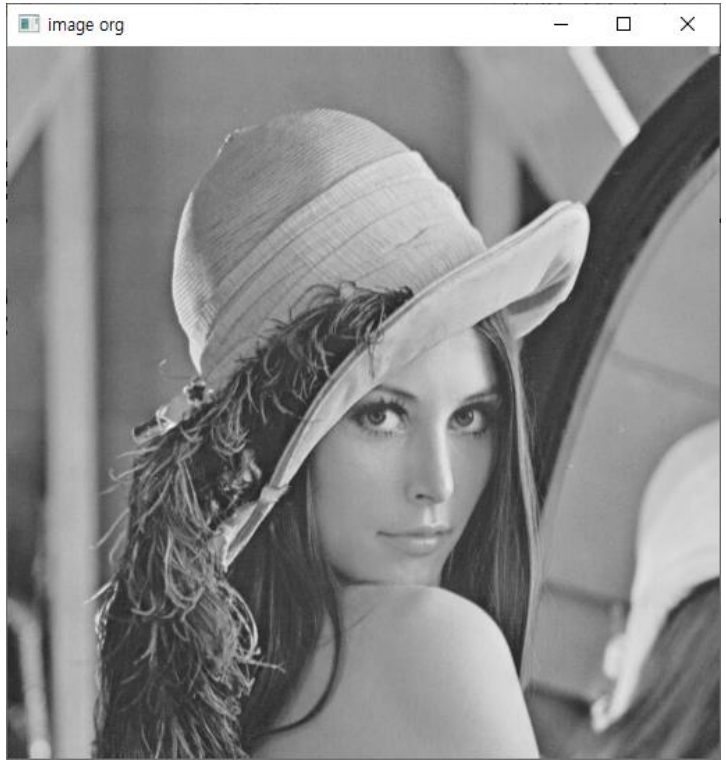
dtype = np.float32



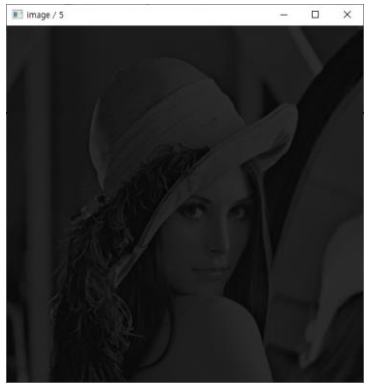
dtype = np.float32

실습(IP2_2)

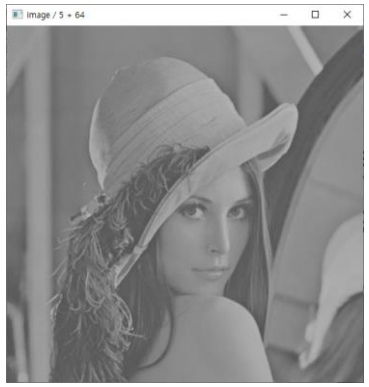
- 영상 변환 연습



original



Original / 5



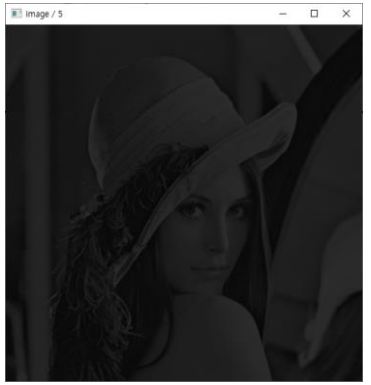
Original / 5 + 64

Original * 2

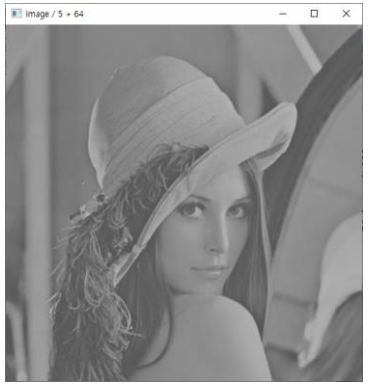
실습(IP2_2)

- 영상 변환 연습

```
def main():
    img_org = cv2.imread('lena.png', cv2.IMREAD_GRAYSCALE)
    img_div5 = img_org.astype(np.float32)
    img_div5 = img_div5 / 5
    #img_div5 = cv2.divide(img_div5, 5)
    img_div5 = np.clip(img_div5, 0, 255)

    img_mul2 = img_org.astype(np.float32)
    img_mul2 = img_mul2 * 2
    #img_mul2 = cv2.multiply(img_mul2, 2)
    img_mul2 = np.clip(img_mul2, 0, 255)

    img_trunc = img_org.astype(np.float32)
    img_trunc = img_trunc / 2 + 64
    img_trunc = np.clip(img_trunc, 0, 255)

    cv2.imshow('image org', img_org)
    cv2.imshow('image / 5', img_div5.astype(np.uint8))
    cv2.imshow('image * 2', img_mul2.astype(np.uint8))
    cv2.imshow('image / 5 + 64', img_trunc.astype(np.uint8))
    cv2.waitKey()
    cv2.destroyAllWindows()
```



Original / 5

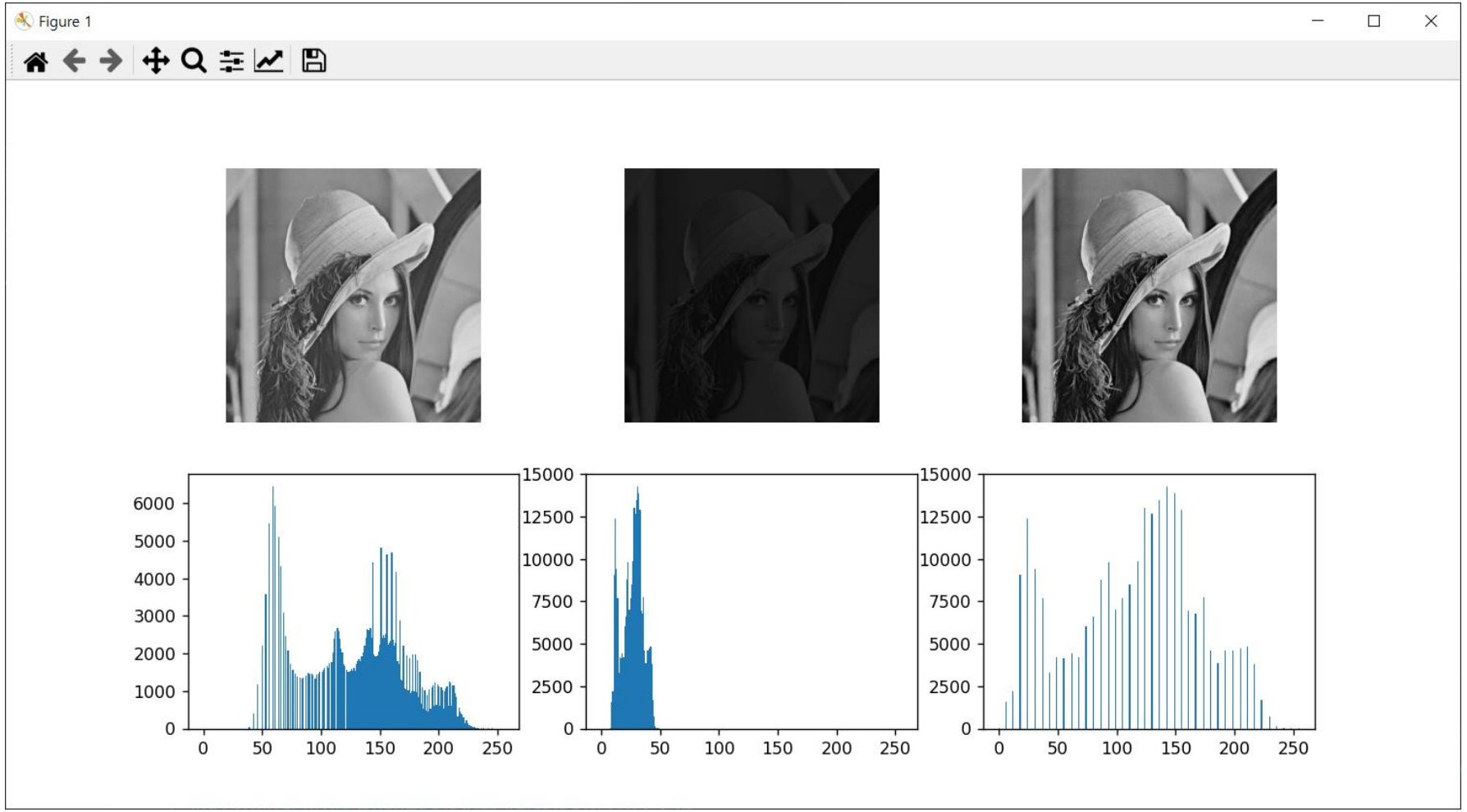


Original / 5 + 64

Original * 2

과제(IP2_test1)

- 히스토그램 그리기 & stretching



과제(IP2_test1)

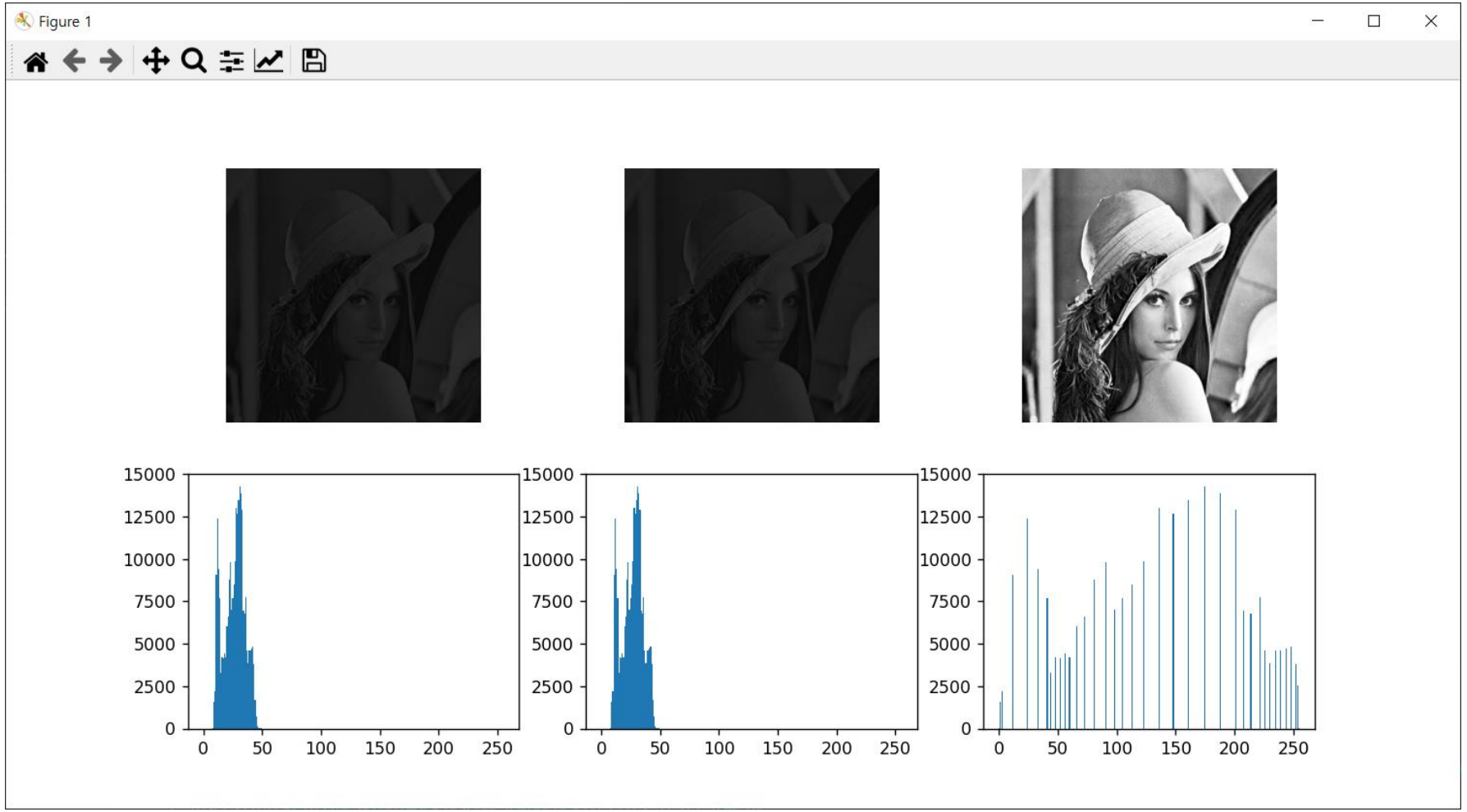
- 히스토그램 그리기 & stretching

```
def getHistogram(img):
    """
    [input]
    img : image

    [output]
    hist : histogram 1D array ( shape : (256,) )
    """
    return hist
```

```
def histStretching(img):
    """
    [input]
    img : image

    [output]
    img : stretched image
    """
    return img
```

과제(IP2_test2)

- Histogram equalization



과제(IP2_test2)

- Histogram equalization

```python
def histEqualization(img, max_intensity = 255):
    # calculate histogram
    hist = getHistogram(img, max_intensity)

    # calculate cumulative histogram
    # don't use np.cumsum
    hist_cum = cumsum(hist)

    # divide cumulative value
    hist_norm = 'ToDo'

    # equalize histogram
    hist_equal = 'ToDo'

    # apply equalization
    img_equal = 'ToDo'

    return img_equal, hist_equal
```

```python
def cumsum(hist):
    """
    [input]
    hist : histogram 1D array ( shape : (256,) )

    [output]
    hist_c : Accumulated histogram 1D array ( shape : (256,) )
    """

    return hist_c
```

```python
def getHistogram(img, max_intensity=255):
    """
    [input]
    img : image
    max_intensity : max intensity( default : 255 )

    [output]
    hist : histogram 1D array ( shape : (256,) )
    """

    return hist
```

QnA