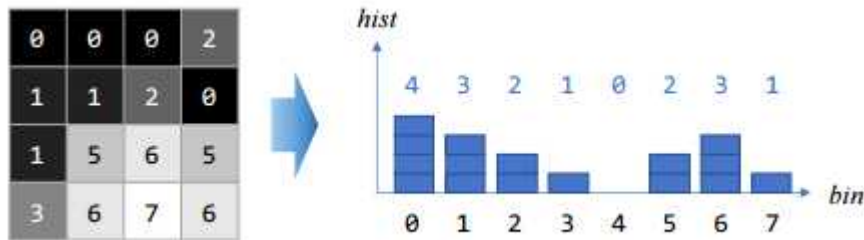


디지털 영상처리 연구실 연구보고서

정지우

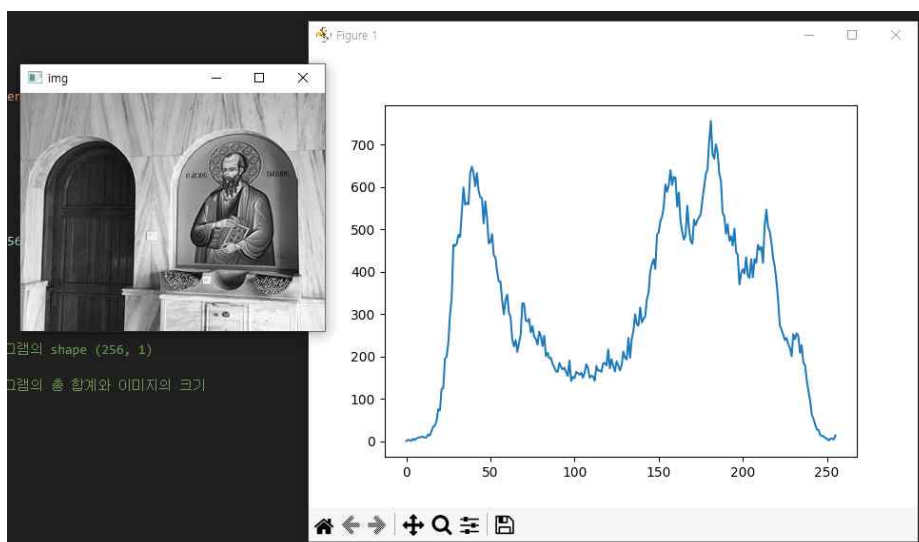
히스토그램(Histogram)은 영상의 픽셀 값 분포를 그래프의 형태로 표현한 것



이러한 분포를 통해 영상의 특성을 판단할 수 있는 도구로 활용된다.

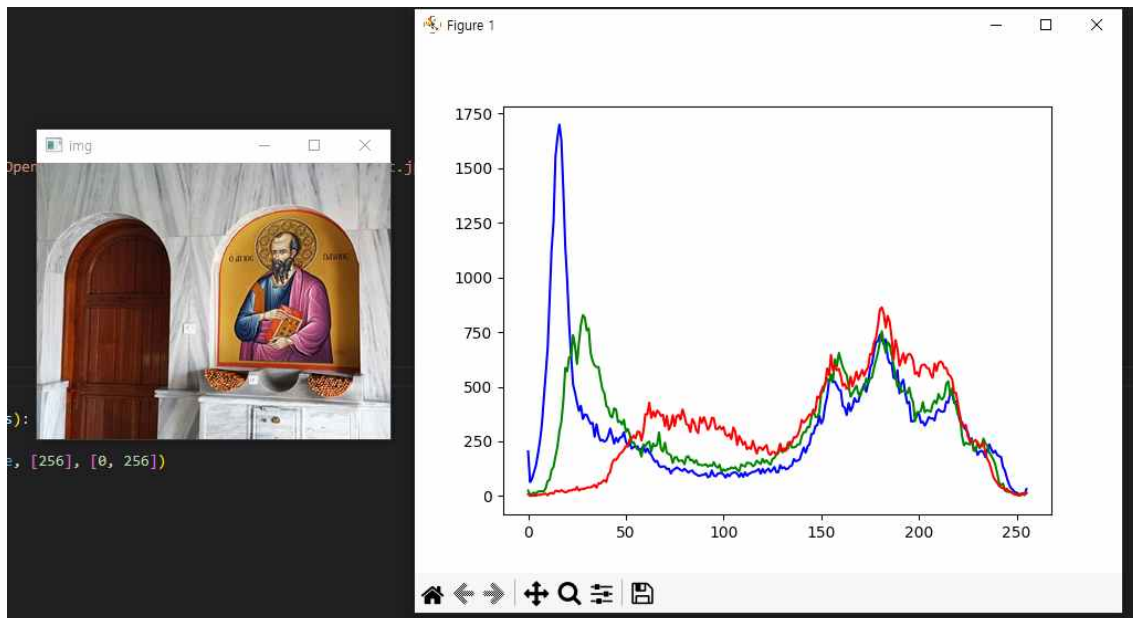
```
1 import numpy as np, cv2
2
3 import matplotlib.pyplot as plt
4
5
6 |
7 # 이미지를 그레이 스케일로 읽기 및 출력
8
9 img = cv2.imread("C:/Users/PC/Desktop/OpenCV (Python)/source/chap06/images/draw_hist.jpg", cv2.IMREAD_GRAYSCALE)
10
11 cv2.imshow('img', img)
12
13
14 # 히스토그램 계산 및 그리기
15
16 hist = cv2.calcHist([img], [0], None, [256], [0, 256])
17
18 plt.plot(hist)
19
20
21
22
23 print(hist.shape) # 히스토그램의 shape (256, 1)
24
25 print(hist.sum(), img.shape) # 히스토그램의 총 합계와 이미지의 크기
26
27 plt.show()
```

Matplotlib은 Python 프로그래밍 언어 및 수학적 확장 NumPy 라이브러리

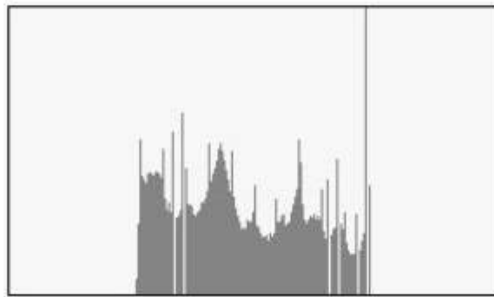


컬러 히스토그램

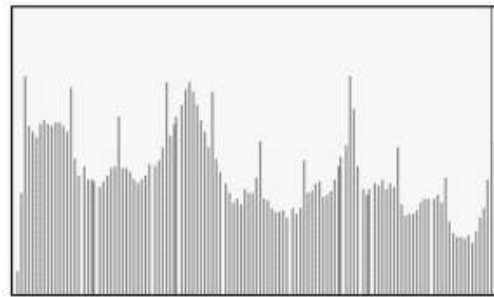
```
1 import numpy as np, cv2
2
3 import matplotlib.pyplot as plt
4
5
6
7 # 이미지를 읽기 및 출력
8
9 img = cv2.imread("C:/Users/PC/Desktop/OpenCV (Python)/source/chap06/images/draw_hist.jpg")
10
11 cv2.imshow('img', img)
12
13
14
15 # 히스토그램 각각 계산 및 그리기
16
17 channels = cv2.split(img)
18
19 colors = ['b', 'g', 'r']
20
21 for (ch, color) in zip(channels, colors):
22
23     hist = cv2.calcHist([ch], [0], None, [256], [0, 256])
24
25     plt.plot(hist, color = color)
26
27
28
29 plt.show()
```



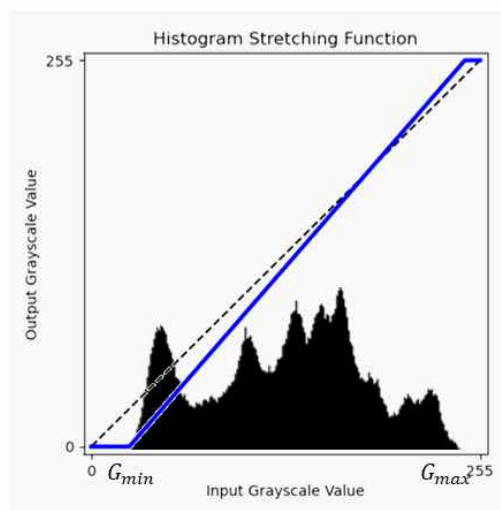
스트레칭



스트레칭 이전



스트레칭 이후



• 수식

$$y = ax + b$$

$$a = \frac{255 - 0}{G_{max} - G_{min}}$$

$$y = \frac{255}{G_{max} - G_{min}}(x - G_{min})$$

$$\therefore dst(x, y) = \frac{src(x, y) - G_{min}}{G_{max} - G_{min}} \times 255$$

```

# 직접 연산한 정규화

img_f = img.astype(np.float32)

img_norm = ((img_f - img_f.min()) * (255) / (img_f.max() - img_f.min()))

img_norm = img_norm.astype(np.uint8)

# OpenCV API를 이용한 정규화

img_norm2 = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX)

# 히스토그램 계산

hist = cv2.calcHist([img], [0], None, [256], [0, 255])

hist_norm = cv2.calcHist([img_norm], [0], None, [256], [0, 255])

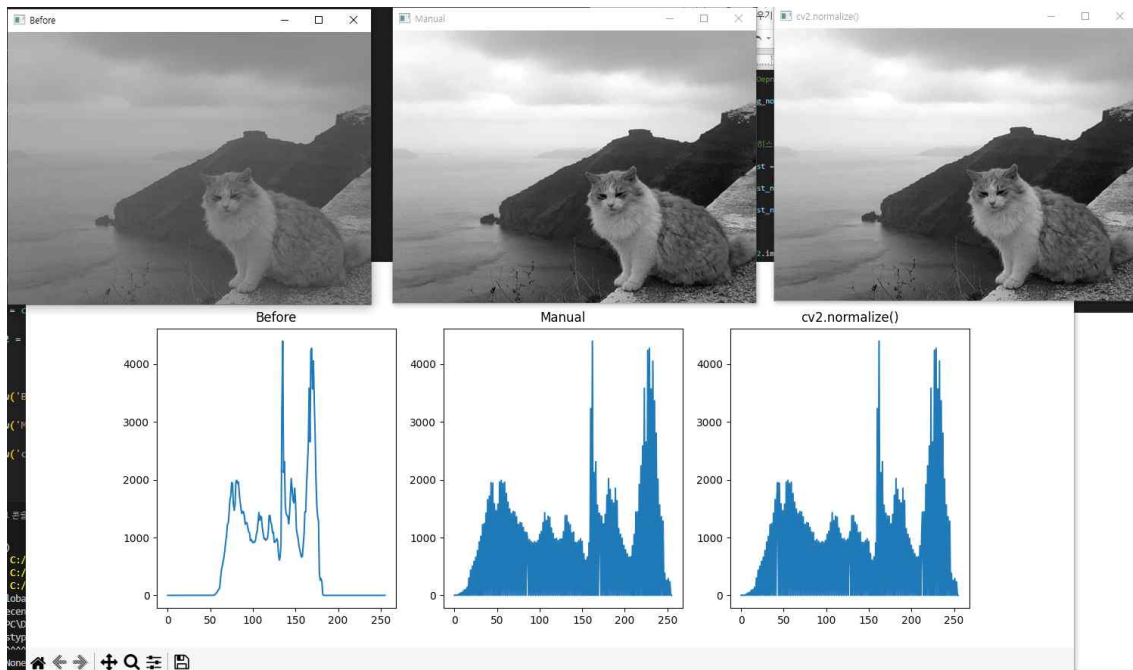
hist_norm2 = cv2.calcHist([img_norm2], [0], None, [256], [0, 255])

cv2.imshow('Before', img)

cv2.imshow('Manual', img_norm)

cv2.imshow('cv2.normalize()', img_norm2)

```



평활화

이미지의 밝기 값 분포를 고르게 하여 더 넓은 범위의 밝기 값에 걸쳐 분포하도록 만든다.

히스토그램 평활화를 위한 변환 함수 구하기

- 히스토그램 함수 구하기 : $h(g) = N_g$
- 정규화된 히스토그램 함수 구하기 : $p(g) = \frac{h(g)}{w \times h}$
- 누적 분포 함수(cdf) 구하기 : $cdf(g) = \sum_{0 \leq i \leq g} p(i)$
- 변환 함수 : `dst(x, y) = round(cdf(src(x, y)) x Lmax`

```
13 # OpenCV API로 이퀄라이즈 히스토그램 적용
14
15 img2 = cv2.equalizeHist(img)
16
17
18
19 # 이퀄라이즈 결과 히스토그램 계산
20
21 hist = cv2.calcHist([img], [0], None, [256], [0, 256])
22
23 hist2 = cv2.calcHist([img2], [0], None, [256], [0, 256])
24
25
26
27 # 결과 출력
28
29 cv2.imshow('Before', img)
30
31 cv2.imshow('cv2.equalizeHist()', img2)
32
33 hists = {'Before': hist, 'cv2.equalizeHist': hist2}
34
35 for i, (k, v) in enumerate(hists.items()):
36
37     plt.subplot(1, 2, i+1)
38
39     plt.title(k)
40
41     plt.plot(v)
42
43
44
45 plt.show()
```

