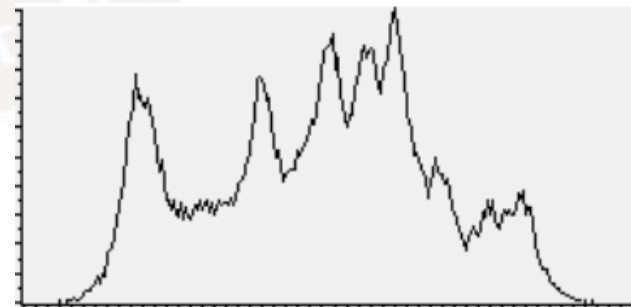
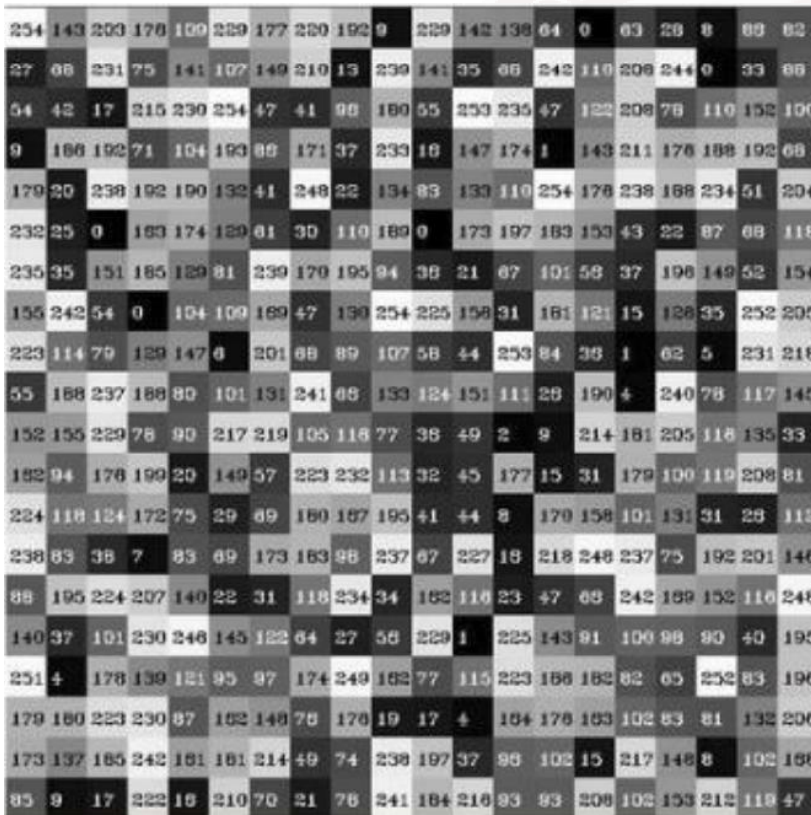


1-2. 히스토그램과 영상 이진화

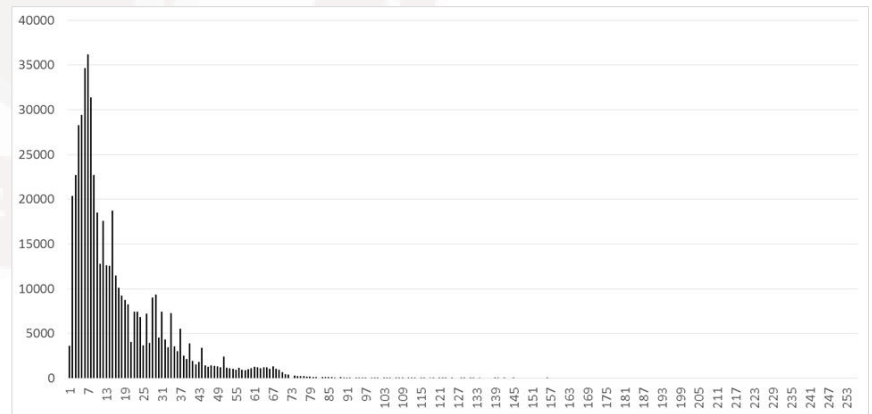
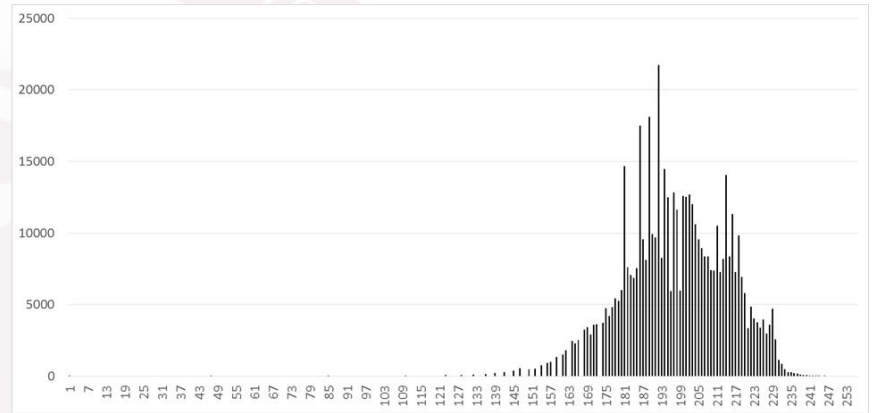
■ 히스토그램(histogram)

- 영상 히스토그램이란 영상에 사용된 밝기 값들의 분포를 그래프로 표현하는 것
- 밝기 값들의 분포는 영상 내에 존재하는 화소들의 밝기 값의 빈도수를 의미함
- 특징: 밝기 값의 분포에 대한 분석을 통하여, 영상 전체에 있어서 색의 밝고 어두운 것을 분석 가능



■ 히스토그램(histogram)

- 각 화소들은 0에서 255 사이의 한 값으로 표현 가능
- x 축이 0~255의 값들을 의미하며 칸(bin)이라고 부름



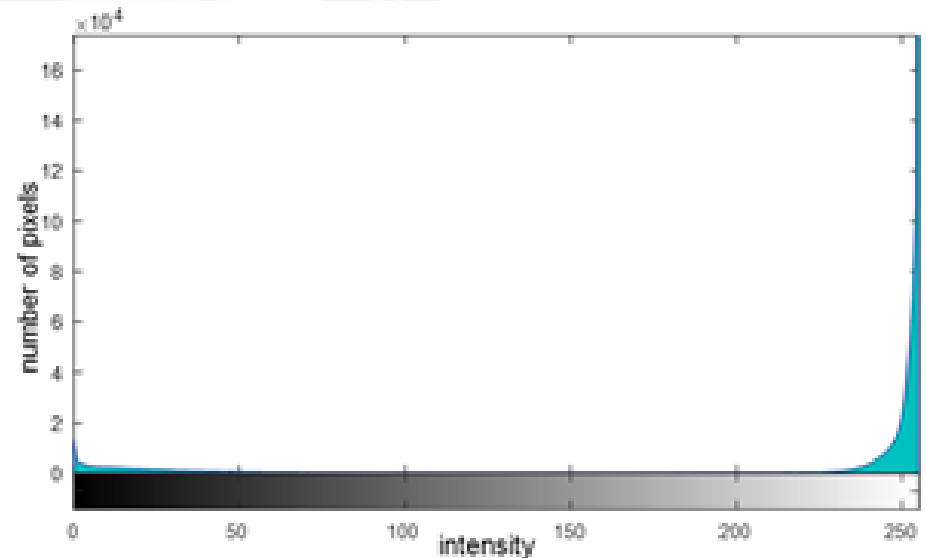
1.4 히스토그램

■ 히스토그램(histogram)

- 각 화소들은 0에서 255 사이의 한 값으로 표현 가능
- x 축이 0~255의 값들을 의미하며 칸(bin)이라고 부름

6.2.5 변환계수를 이용한 특징 벡터

변환 계수를 이용한 특징값을 구하는 방법도 넓은 의미에서 통계적 특징값으로 고려될 수 있을 것이다. 입력 영상의 공간 평면 데이터를 주파수 평면, Haar 평면, DCT 평면과 같은 또 다른 기저함수 평면으로 변환하여, 그 계수 값을 특징 벡터로 사용할 수 있을 것이다. 여기서, 어떠한 변환을 사용하느냐는 것은 응용과 입력 데이터의 특성에 의존적일 것이다. 경우에 따라서, 입력 데이터의 통계적 특성을 고려한 Principle component analysis (PCA)를 이용하여, 기저 벡터로의 투영 분산값이 최대가 되도록 최적의 기저 벡터를 구할 수 있다. 이렇게 구한 기저 벡터를 이용한 변환 계수를 이용하여, 특징 벡터를 생성할 수 있고, 이러한 벡터 값을 넓은 의미에서 통계적 특징 벡터라고 볼 수도 있겠다. 그림 6은 8x8 영상 데이터에 대한 DCT 변환 기저함수 64개를 보여주고, 입력 신호와 기저함수를 내적함으로써 DCT 계수를 구할 수 있을 것이다. 입력 영상 밝기값 자체를 사용하는 것이 아니고, 입력 영상에 DCT 기저 함수 등 또 다른 기저함수를 이용하여 변환계수 벡터를 구할 수 있고, 이것을 통계적 특징으로 해석할 수 있겠다.



■ 이진화란?

- 화소가 0(검정) 또는 1(흰색)인 영상
- 1비트로 저장할 수 있으나, 편의상 1바이트 사용하는 경우가 많음
- 에지 검출 결과를 표시하거나 물체와 배경을 구분하여 표시하는 응용 등에 사용



■ 이진화 어떻게?

- 명암 영상을 이진화하려면 임계값 T 보다 큰 화소는 1, 그렇지 않은 화소는 0으로 변환

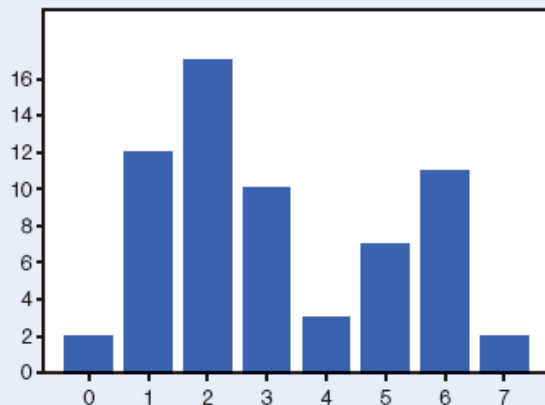
$$b(x, y) = \begin{cases} 1, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases}$$

1	2	2	2	1	1	2	0
2	6	7	6	6	4	3	0
2	6	7	6	6	4	3	2
2	5	6	6	6	4	3	2
2	5	6	6	5	5	3	2
2	5	5	5	3	3	3	2
2	2	3	3	3	1	1	1
2	2	1	1	1	1	1	1

(a) 입력 영상

$h =$

2	12	17	10	3	7	11	2
---	----	----	----	---	---	----	---



(b) 히스토그램

0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	0
0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(c) 이진 영상

■ 이진화 코드: threshold() 함수

- threshold 함수는 입력한 영상 데이터에 대해 임계값 필터링을 수행하여 이진영상을 생성

함수명	cv2.threshold(src, thresh, maxval, type)
매개변수	<ul style="list-style-type: none">- src (numpy.ndarray) 입력 영상 데이터- thresh 임계값 (threshold value)- maxval THRESH_BINARY & THRESH_BINARY_INV에서 사용할 최대값- type 임계값 필터링 종류 (Threshold Type)
리턴값	출력 영상 데이터 (numpy.ndarray)

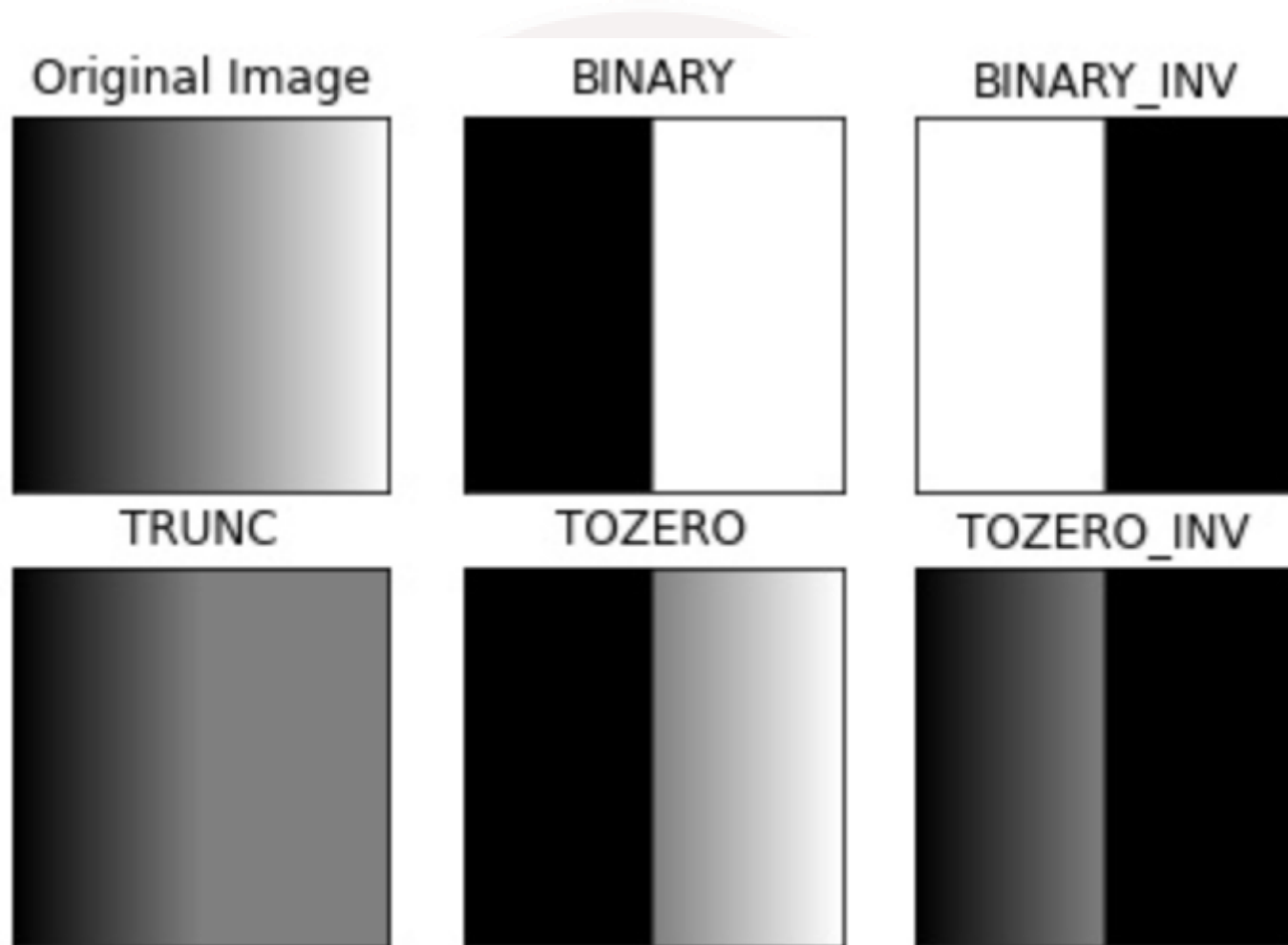
■ 이진화 코드: threshold() 함수

- threshold 함수는 입력한 영상 데이터에 대해 임계값 필터링을 수행하여 이진영상을 생성

type	THRESH_BINARY Python: cv.THRESH_BINARY	$\text{dst}(x, y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$
	THRESH_BINARY_INV Python: cv.THRESH_BINARY_INV	$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$
	THRESH_TRUNC Python: cv.THRESH_TRUNC	$\text{dst}(x, y) = \begin{cases} \text{threshold} & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$
	THRESH_TOZERO Python: cv.THRESH_TOZERO	$\text{dst}(x, y) = \begin{cases} \text{src}(x, y) & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$
	THRESH_TOZERO_INV Python: cv.THRESH_TOZERO_INV	$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$

■ 이진화 코드: threshold() 함수

- threshold 함수는 입력한 영상 데이터에 대해 임계값 필터링을 수행하여 이진영상을 생성



■ 이진화 코드

```
import cv2

src = cv2.imread('./images/Lena_2.jpg', cv2.IMREAD_GRAYSCALE)
cv2.imshow('src', src)

_, dst = cv2.threshold(src, 160, 255, cv2.THRESH_BINARY)
cv2.imshow('dst', dst)

cv2.waitKey(0)
```

