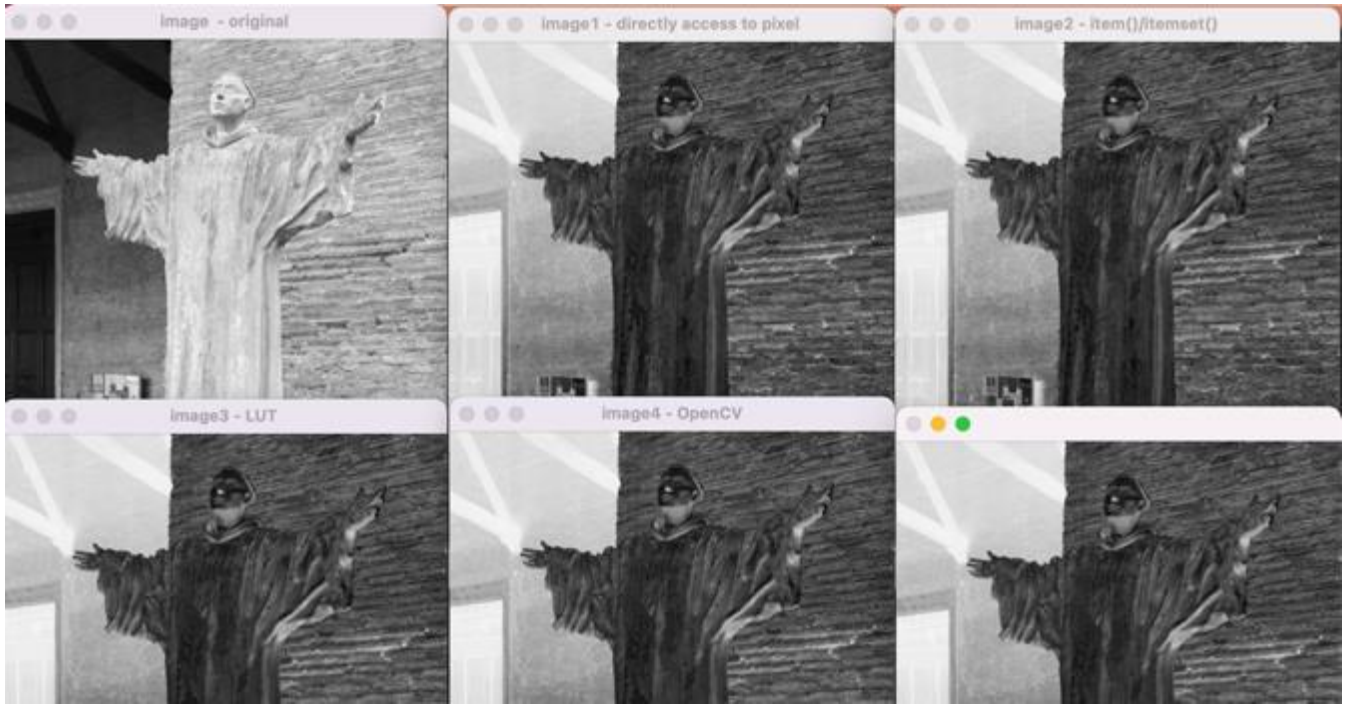
화소(pixel) 처리

영상 화소 접근

- 픽셀(행렬 원소) 접근
  - ✓ 영상의 반전 – 다양한 방법을 함수로 생성하고 수행 속도를 계산



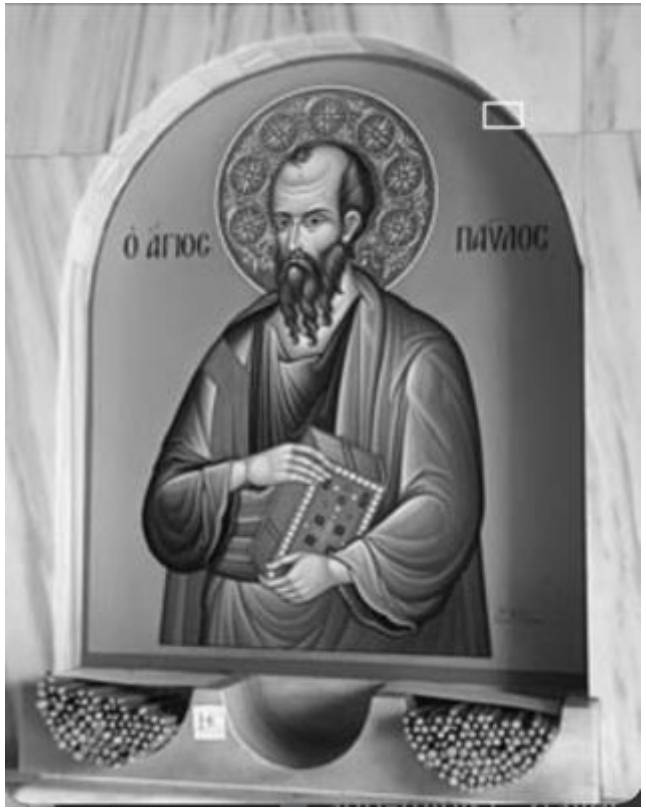
영상 화소 밝기 변환

- 그레이 스케일 영상
  - ✓ 단일 채널의 영상 즉 일반적으로 이해하는 컬러가 아닌 영상을 흑백 영상이라고 하는데 흑백 영상이라는 것은 검은색과 흰색으로 구성된 영상을 의미
  - ✓ 디지털 영상처리에서 보통 단일 채널의 영상을 그레이 스케일(gray-scale) 영상이라고 하기도 하고 명암도 영상이라 함
  - ✓ 픽셀 값의 그레이 스케일 표현

  - ✓ 하나의 픽셀 값은 0~255의 값을 가지는데 0은 검은색을 255는 흰색을 의미
  - ✓ 0~255 사이의 값들은 진한 회색에서 연한 회색까지를 나타냄
  - ✓ 픽셀 값이 회색의 비율 정도로 표현되고 0~255의 값을 가지는 픽셀들이 모여서 구성된 것이 영상이기 때문에 그레이 스케일 영상이라고도 함

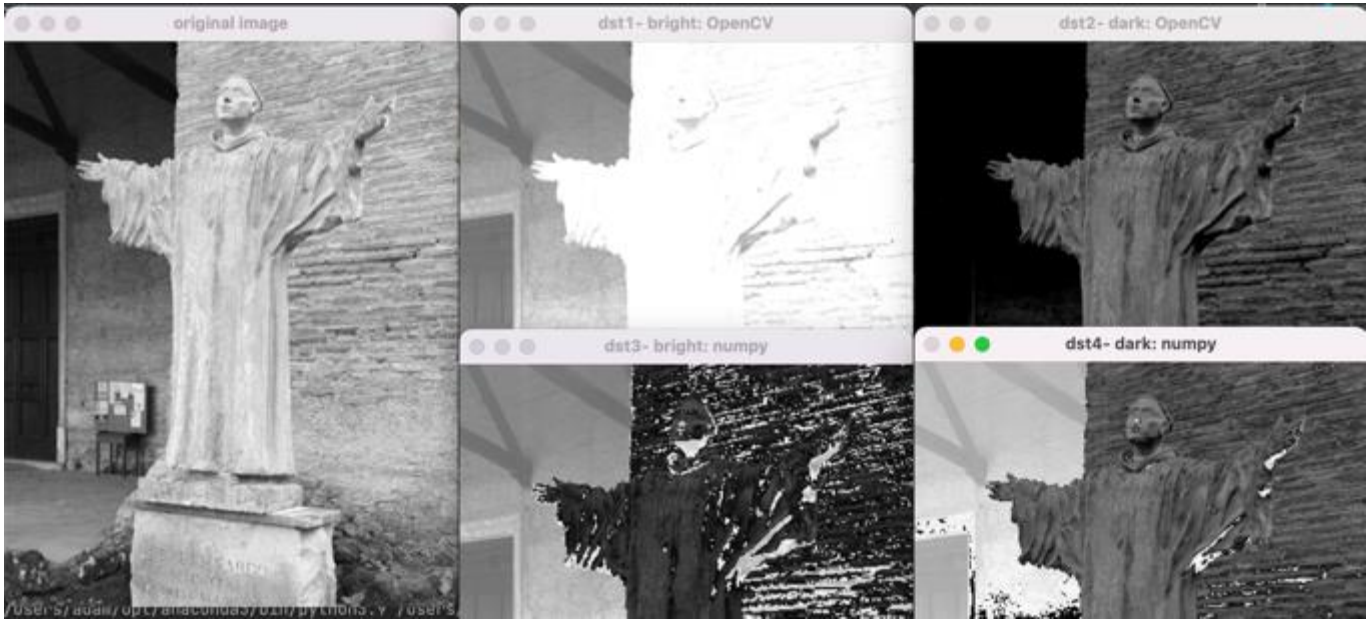
영상 화소 밝기 변환

- 영상의 픽셀 표현
  - ✓ 영상 파일을 행렬에 저장하고 관심 영역을 지정해서 출력하면 간단하게 영상 데이터인 픽셀들의 값을 출력할 수 있음
  - ✓ 영상 픽셀 값 확인



영상 화소 밝기 변환

- ❖ 흑백 영상 밝기의 가감 연산
  - ✓ 픽셀 값이 영상의 밝기를 나타내기 때문에 이 픽셀 값을 변경하면 영상의 밝기를 바꿀 수 있음
  - ✓ 영상의 픽셀에 특정한 상수 값을 더하면 영상이 밝아지고 상수 값을 빼면 영상이 어두워짐
  - ✓ 픽셀이 가질 수 있는 최댓값(255)에서 그 픽셀의 값을 빼면 반전 영상이 만들어짐
  - ✓ 행렬의 덧셈과 뺄셈은 OpenCV의 saturation 방식과 numpy의 modulo 방식이 있음
  - ✓ 영상의 밝기 변경



영상 화소 밝기 변환

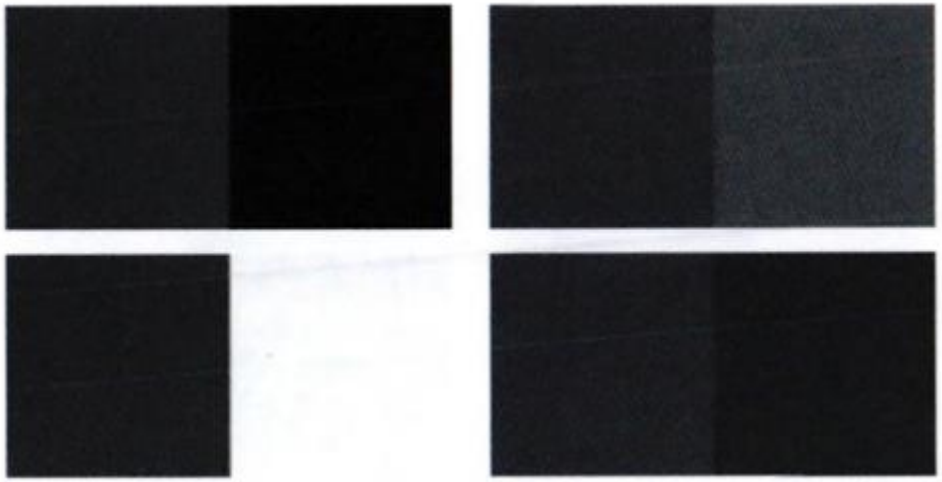
- 영상 덧셈 및 곱셈을 이용한 영상 합성
  - ✓ 두 영상을 합하면 영상 합성이 되며 두 영상을 빼면 차영상(difference image)
  - ✓ 두 개의 행렬을 합하게 되면 saturation 연산으로 인해서 255를 넘어서는 픽셀들은 흰색으로 나타나서 영상의 합성이 제대로 수행되지 않는데 이 문제에 대한 해결 방법은 여러 가지가 있을 수 있음
    - dst(y,x) = image1(y,x) * 0.5 + image2(y,x) * 0.5 ;
    - dst(y,x) = image1(y,x) * alpha + image2(y,x) * (1-alpha)
    - dst(y,x) = image1(y,x) * alpha + image2(y,x) * b

영상 화소 밝기 변환

- ❖ 영상 덧셈 및 곱셈을 이용한 영상 합성
  - ✓ 행렬 합 과 곱 연산을 통한 영상 합성

영상 화소 밝기 변환

- 명암 대비
  - ✓ 명암 대비(contrast)는 상이한 두 가지 색(밝기)이 경계에서 서로 영향을 미쳐 그 차이가 강조되어 나타나는 현상
  - ✓ 다음 그림은 명암 대비의 예를 보인 것으로 각 그림에서 왼쪽 부분은 픽셀 값이 100으로 동일하지만 오른쪽 부분의 다른 밝기로 인해 경계 부분에서 약간 다른 색으로 인식됨

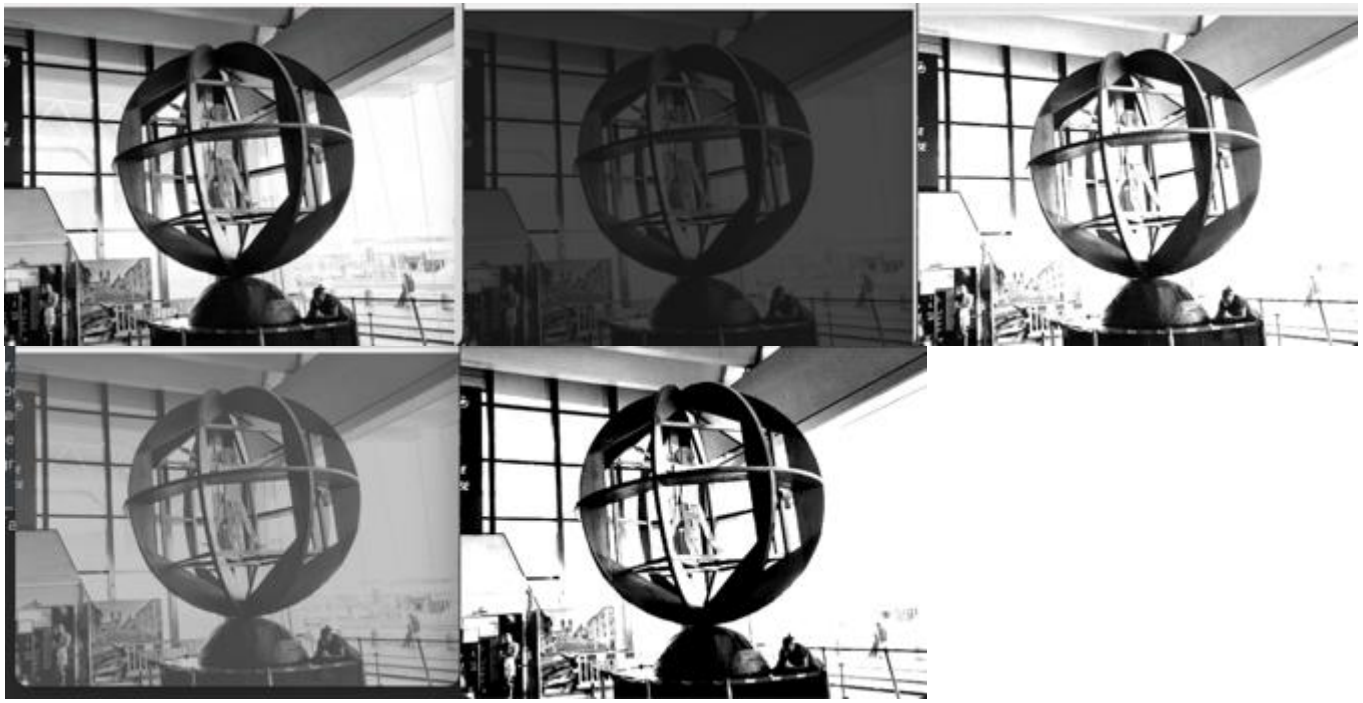


영상 화소 밝기 변환

- 명암 대비
  - ✓ 낮은 명암 대비의 영상은 밝은 부분과 어두운 부분의 차이가 크지 않아 전체적으로 어둡거나 밝은 영상
  - ✓ 높은 명암 대비의 영상은 밝은 부분과 어두운 부분의 차이가 큰 영상을 의미하는데 높은 명암 대비 영상은 어두운 부분과 밝은 부분이 대비를 이루어 전체적으로 영상이 또렷해 보임

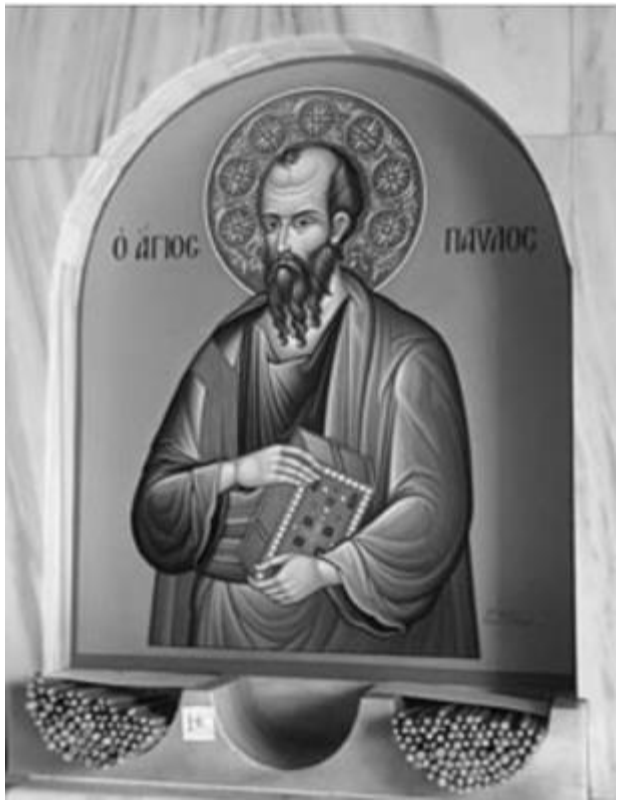
영상 화소 밝기 변환

- 명암 대비
  - ✓ 명암 대비를 높이려면 어두운 부분은 더 어둡게 그리고 밝은 부분은 더 밝게 해야 함
  - ✓ 명암 대비를 낮추려면 어두운 부분과 밝은 부분의 차이를 작게 만들어야 함
  - ✓ 영상 내에서 큰 값들과 작은 값의 차이를 늘리거나 줄이는 쉬운 방법은 곱셈 연산을 수행하면 되는데 명암 대비를 늘리기 위해서는 1.0 이상의 값을 곱해주면 되고 줄이기 위해서는 1.0 이하의 값을 곱해주면 됨
  - ✓ 영상 대비 변경



히스토그램

- Histogram
  - ✓ Histogram(histogram)은 관측값의 개수를 겹치지 않는 다양한 계급으로 표시하는 것
  - ✓ 단일 채널 8비트(uint8) 행렬의 Histogram은 부호 없는 8비트 행렬이기에 나타날 수 있는 원소 값의 범위는 0~255이며 한 계급(bin)이 1이 되게 설정하는데 계급 간격은 경우에 따라 조절
  - ✓ 직접 계산



히스토그램

- Histogram
  - ✓ 다채널의 행렬에서 Histogram 계산

    cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]) -> hist

    - images는 Histogram을 계산할 영상의 배열로 영상은 같은 크기로 같은 깊이의 8비트 부호 없는 정수 또는 32비트 실수 자료형
    - channels는 Histogram을 계산할 채널 번호를 갖는 정수형 배열
    - mask는 images[i]와 같은 크기의 8비트 영상으로, mask(x, y) != 0인 image[i](x, y)만을 Histogram 계산에 사용하는데 mask = None이면 마스크를 사용하지 모든 픽셀에서 Histogram을 계산
    - histSize는 결과 Histogram hist의 각 bin 크기에 대한 정수 배열
    - ranges는 Histogram의 각 빈의 경계값 배열의 배열로 OpenCV 파이썬은 등간격 Histogram을 계산
    - accumulate = True이면 calcHist() 함수를 수행할 때 Histogram을 초기화하지 않고 이전 값을 계속 누적
    - hist는 결과 Histogram

히스토그램

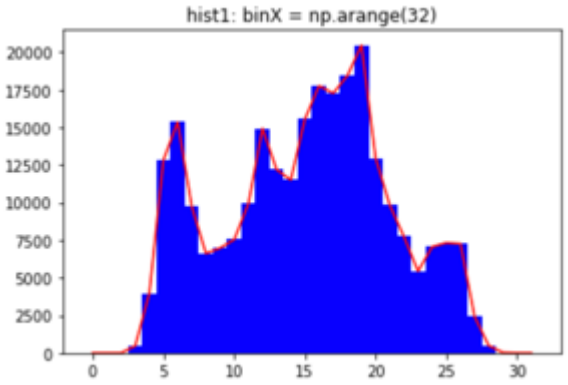
- Histogram
  - ✓ Color Histogram
    - ❑ hist2 = cv2.calcHist(images = [src], channels = [0], mask = None, histSize = [4], ranges = [0, 4])는 4 X 4 배열 src의 0번 채널에서 마스크 지정없이 Histogram 빈 크기 histSize = [4], 범위 ranges = [0, 4]로 Histogram hist2를 계산하는데 범위 [0, 4]에서 0은 포함 4은 포함하지 않음
      - hist1[0][0] = 4는 src에서 0 의 카운트
      - hist1[1][0] = 5는 src에서 1 의 카운트
      - hist1[2][0] = 0는 src에서 2 의 카운트
      - hist1[3][0] = 3는 src에서 3 의 카운트

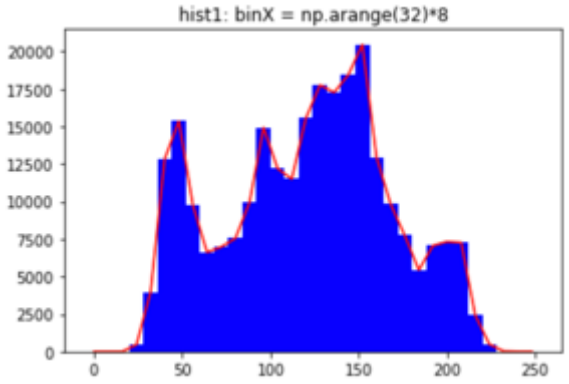
히스토그램

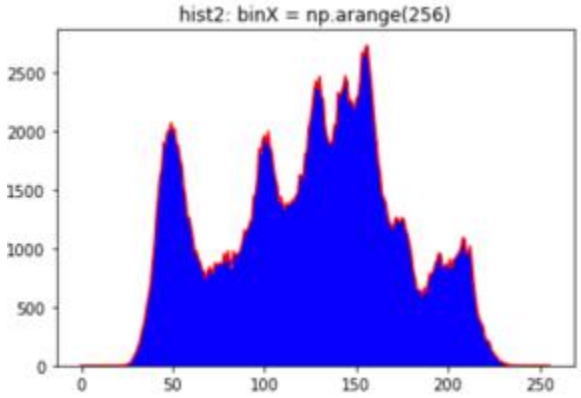
- ❖ Histogram
  - ✓ 흑백 이미지 Histogram
    - ❑ hist1 = cv2.calcHist(images = [src], channels = [0], mask = None, histSize = [32], ranges = [0, 256])는 그레이스케일 영상 src의 0번 채널에서 마스크 지정 없이, Histogram 빈 크기 histSize = [32], 범위 ranges = [0, 256]으로 Histogram hist2을 계산
      - hist1[0][0]은 영상 src에서 0 에서 31 까지의 카운트
      - hist1[1][0]은 영상 src에서 32 에서 63 까지의 카운트
      - …
      - hist1[31][0]은 영상 src에서 224 에서 255 까지의 카운트
    - ❑ hist2 = cv2.calcHist(images = [src], channels = [0], mask = None, histSize = [256], ranges = [0, 256])는 영상 src의 0번 채널에서 마스크 지정 없이, Histogram 빈 크기 histSize = [256], 범위 ranges = [0, 256]으로 Histogram hist2을 계산
      - hist1[0][0]은 영상 src에서 0의 카운트
      - hist1[1][0]은 영상 src에서 1의 카운트
      - …
      - hist1[255][0]은 영상 src에서 255 의 카운트

히스토그램

- ❖ Histogram
  - ✓ 흑백 이미지 Histogram



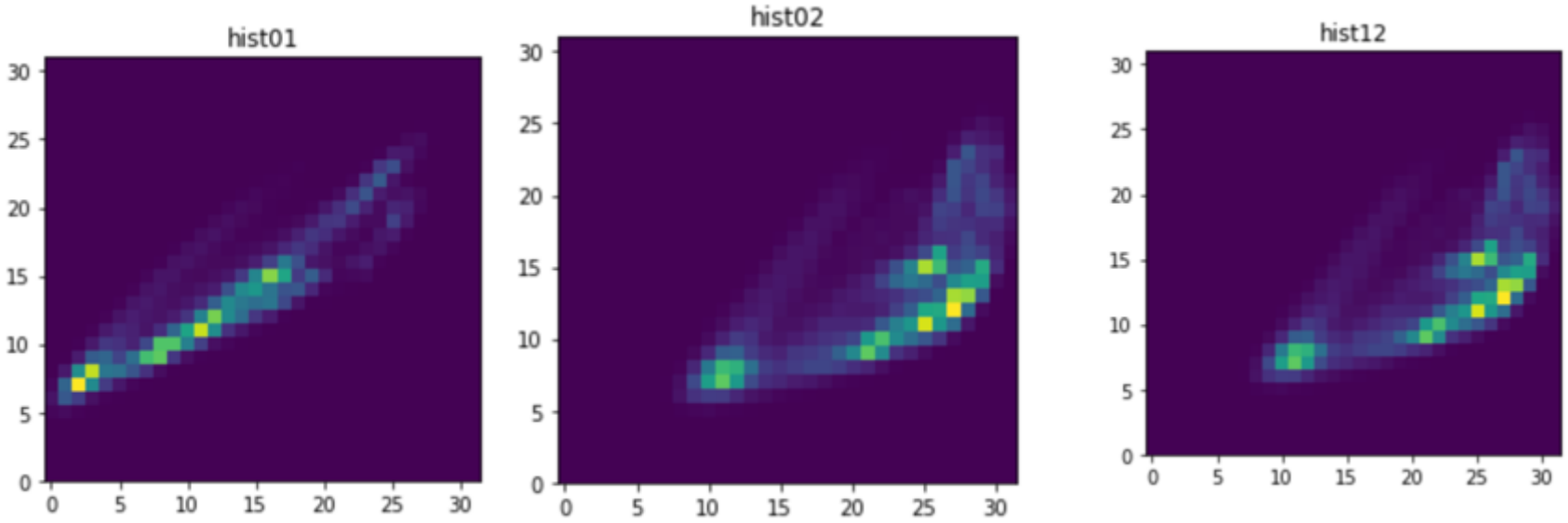




히스토그램

❖ Histogram

✓ 컬러 이미지 2채널 Histogram



❑ #1: cv2.calcHist() 함수로 bgr의 [0, 1] 채널에서 histSize = [32, 32], 범위 ranges = [0, 256, 0, 256]으로 Histogram hist01을 계산하고 plt.imshow()로 표시

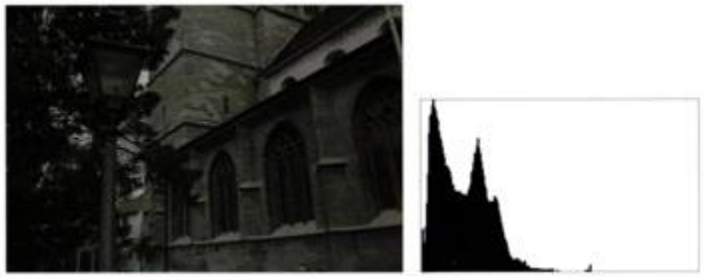
❑ #2: cv2.calcHist() 함수로 bgr의 [0, 2] 채널에서 histSize = [32, 32], 범위 ranges = [0, 256, 0, 256]으로 Histogram hist02를 계산하고 plt.imshow()로 표시

❑ #3 : cv2.calcHist() 함수로 bgr의 [1, 2] 채널에서 histSize = [32, 32], 범위 ranges = [0, 256, 0, 256]으로 Histogram hist12를 계산하고 plt.imshow()로 표시

히스토그램

- Histogram Stretching
  - ✓ 명암도 영상에서 영상이 보기에 선명하고 깨끗해 보이려면 어두운 부분에서 밝은 부분까지 고루 분포되어 있어야 하는데 그렇지 않고 주로 특정 밝기 부분만 있는 영상은 그림에서 보는 바와 같이 전체적으로 선명하지 않으며 또렷하지도 않음
  - ✓ 이런 영상을 히스토그램에서 보면 그림의 오른쪽과 같이 한쪽으로 치우쳐서 히스토그램 분포가 좁은 그래프들로 나타남
  - ✓ 히스토그램의 분포가 좁아서 영상의 대비가 좋지 않은 영상들의 화질을 개선할 수 있는 알고리즘이 히스토그램 스트레칭(histogram stretching)
    - ❑ 밝은 부분이 많이 분포하는 영상

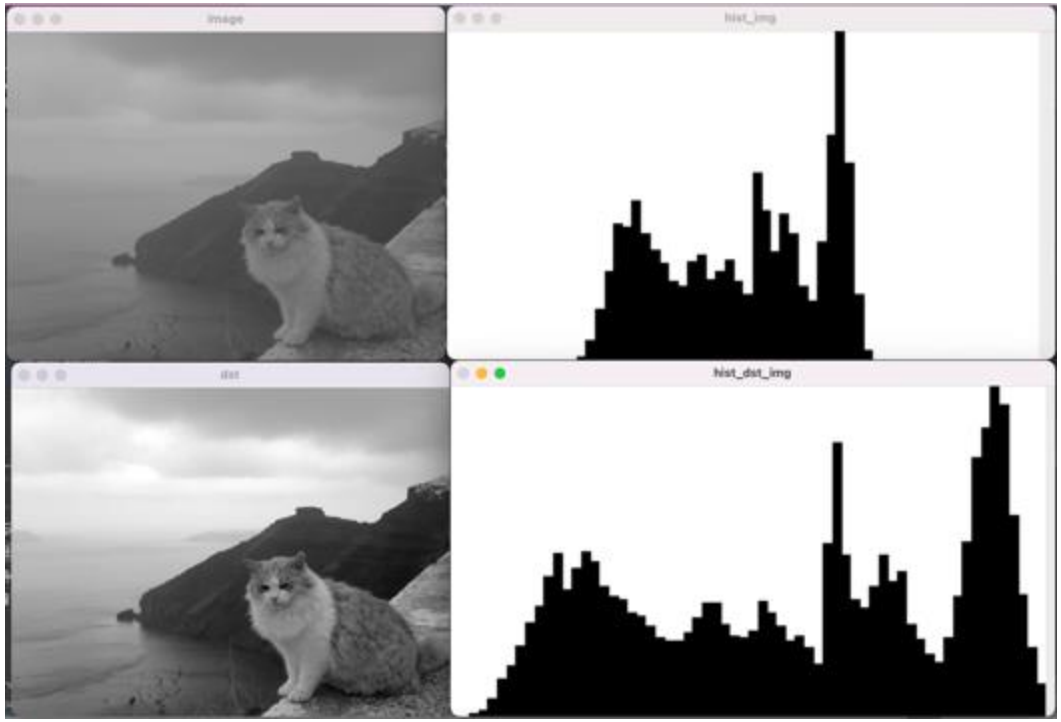
    - ❑ 어두운 부분이 많이 분포하는 영상



히스토그램

- Histogram Stretching
  - ✓ 히스토그램을 스트레칭하고자 하는 경우 빈도 값이 존재하는 가장 낮은 픽셀 값 과 가장 높은 빈도 값이 존재하는 픽셀 값을 알아야 함
  - ✓ 가장 낮은 픽셀 값은 0으로 당기고 가장 높은 픽셀 값을 255 로 당긴 후 중간의 픽셀 값들은 각각의 비율에 따서 픽셀 값의 위치를 조정

    새 픽셀 값 = ((픽셀 값 – low) / (high – low)) * 255



히스토그램

- Histogram Equalization(평활화)
  - ✓ 분포의 균등 이라는 방법을 이용해 명암 대비를 증가시키는데 이를 통해서 영상의 인지도를 높이며 영상의 화질을 개선할 수 있음
  - ✓ 히스토그램 스트레칭은 히스토그램의 분포가 좁은 영상을 스트레칭하여 히스토그램 분포를 넓게 만드는 알고리즘으로 히스토그램의 분포가 좁지는 않지만 특정 부분에서 한쪽으로 치우친 명암 분포를 가진 영상들이 있을 수 있는데 이런 영상들은 명암 분포가 좁지 않기 때문에 히스토그램 스트레칭으로는 문제가 해결되지 않음
  - ✓ 특정 부분에서만 한쪽으로 치우친 명암 분포를 가진 영상을 히스토그램의 재 분배 과정을 거쳐서 균등한 히스토그램 분포를 갖게 하는 알고리즘이 히스토그램 평활화 알고리즘
  - ✓ 히스토그램 평활화를 수행하는 전체 과정
    - ❑ 영상의 히스토그램을 계산
    - ❑ 히스토그램 빈도 값에서 누적 빈도수(누적합)를 계산
    - ❑ 누적 빈도수를 정규화(정규화 누적합)
    - ❑ 결과 픽셀값 = 정규화 누적합 * 최대 픽셀값

히스토그램

❖ Histogram Equalization(평활화)

✓ 평활화 계산 과정

0	2	2	1
1	2	3	2
1	2	3	2
1	3	1	7

입력 영상 화소값

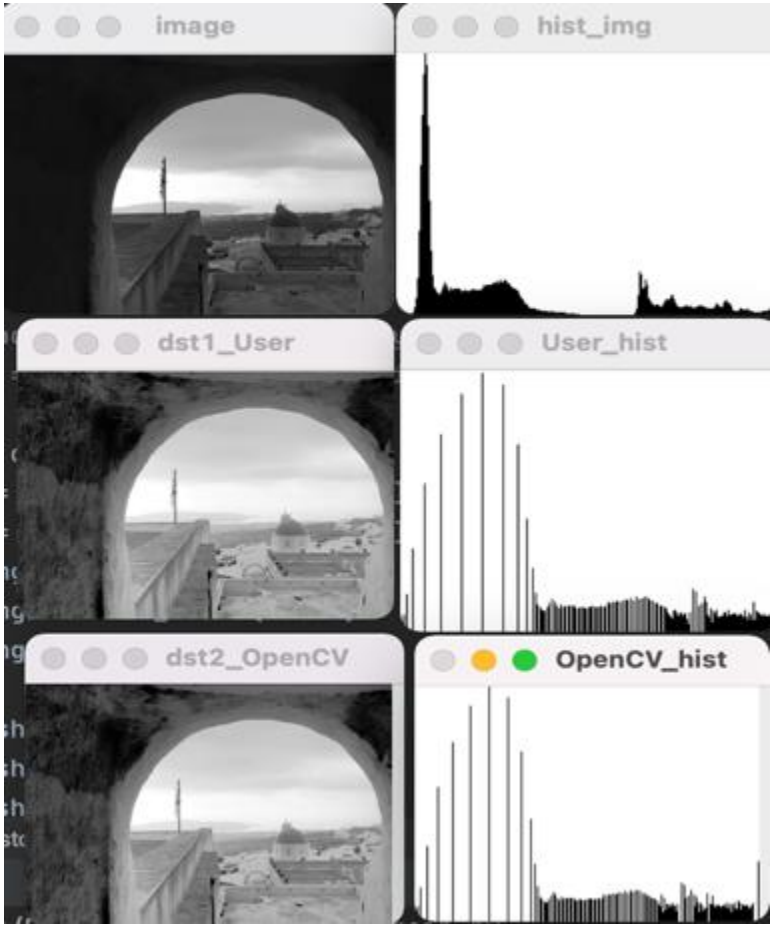
0	5	5	3
3	5	7	5
3	5	7	5
3	7	3	7

평활화 완료 영상 화소값

화소값	0	1	2	3	4	5	6	7
빈도수	1	5	6	3	0	0	0	1
누적 빈도수	1	6	12	15	15	15	15	16
정규화누적합	1/16	6/16	12/16	15//16	15//16	15/16	15/16	16/16
	0.0625	0.375	0.75	0.9375	0.9375	0.9375	0.9375	1
누적합 * 최댓값	0.4375	2.625	5.25	6.5625	6.5625	6.5625	6.5625	7
평활화 결과	0	3	5	7	7	7	7	7

히스토그램

- ❖ Histogram Equalization(평활화)
  - ✓ 평활화



히스토그램

- Histogram Equalization(평활화)
  - ✓ BGR 컬러영상의 Histogram Equalization는 HSV, YCrCb 등의 컬러 모델로 변환한 다음 밝기 값 채널(V, Y)에 Histogram Equalization를 적용한 후에 BGR 컬러영상으로 변환

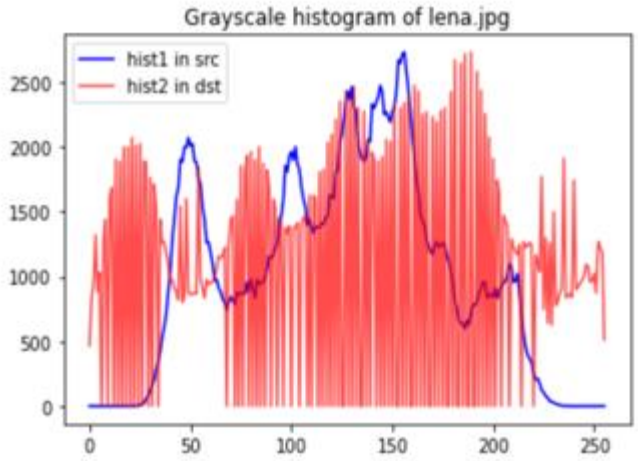
    cv2.equalizeHist(src[, dst]) -> dst

    - ❑ src는 1채널 8비트 입력 영상이고 dst는 src와 같은 크기, 같은 종류의 Histogram Equalization 된 출력 영상
    - ❑ Histogram Equalization 알고리즘
      - src 영상에서 Histogram H를 계산
      - Histogram 빈의 합계가 255가 되도록 정규화하여 H(i) = H(i) X 255 를 생성
      - Histogram 누적 합계. $H'(i) = \sum_{0 \le j \le i} H(j)$를 계산
      - H′을 변환 참조표로 사용하여. dst(x,y) = H'(src(x,y)) 를 계산

        단 dst(x , y) = 0 if src(x , y) = 0

히스토그램

- ❖ Histogram Equalization
  - ✓ 그레이스케일 영상의 Histogram Equalization



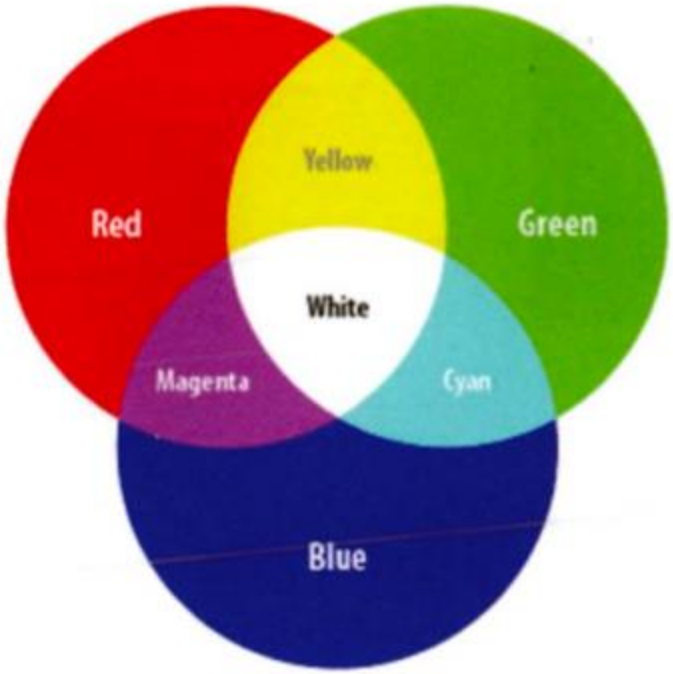
- ❑ 그레이스케일 영상 src를 cv2.equalizeHist()로 Histogram Equalization한 영상 dst는 그림과 같이 선명하게 보임
- ❑ 파란색 꺽은선 그래프는 src의 Histogram이고 빨간색 꺽은선 그래프는 dst의 Histogram인데 빨간색 꺽은선 그래프가 더욱 넓게 분포된 것을 알 수 있음

컬러 공간 변환

- 컬러 및 컬러 공간
  - ✓ 컬러 공간이란 색 표시 계(color system)의 모든 색들을 색 공간에서 3차원 좌표로 표현한 것
  - ✓ 색 표시 계란 RGB, CMY, HSV, LAB, YUV 등의 색 체계를 의미
  - ✓ 컬러 공간을 다른 말로 컬러 표현 시스템(color representation system), 컬러 모델(color model)로도 표현
  - ✓ 컬러 공간은 공간상의 좌표로 표현되기 때문에 어떤 컬러와 다른 컬러들과의 관계를 표현하는 논리적인 방법을 제공
  - ✓ 모니터와 같은 디스플레이 장치에 사용되는 컬러 공간, TV 방송에서 방송 신호를 송출할 때 사용하는 컬러 공간, 프린터에서 인쇄할 때 사용되는 컬러 공간, 그리고 JPEG 압축을 할 때 사용되는 컬러 공간들이 모두 다름
  - ✓ 이러한 컬러 공간들은 서로 변환될 수 있음
  - ✓ 영상 처리에서 컬러 공간은 다양하게 활용되는데 예를 들어 어떤 영상에서 각각의 색상이 다른 객체를 분리하기 위해서 적절한 컬러 공간을 이용해 전체 영상을 컬러 영역별로 분리 할 수 있고 전선의 연결 오류 검사를 위해 기준 색상과 비슷한 색상의 전선인지를 체크하기 위해 사용될 수 있으며 또한 내용 기반 영상 검색에서 색상 정보를 이용하여 원하는 물체를 검색하기 위해 특정 컬러 공간을 이용할 수 있음

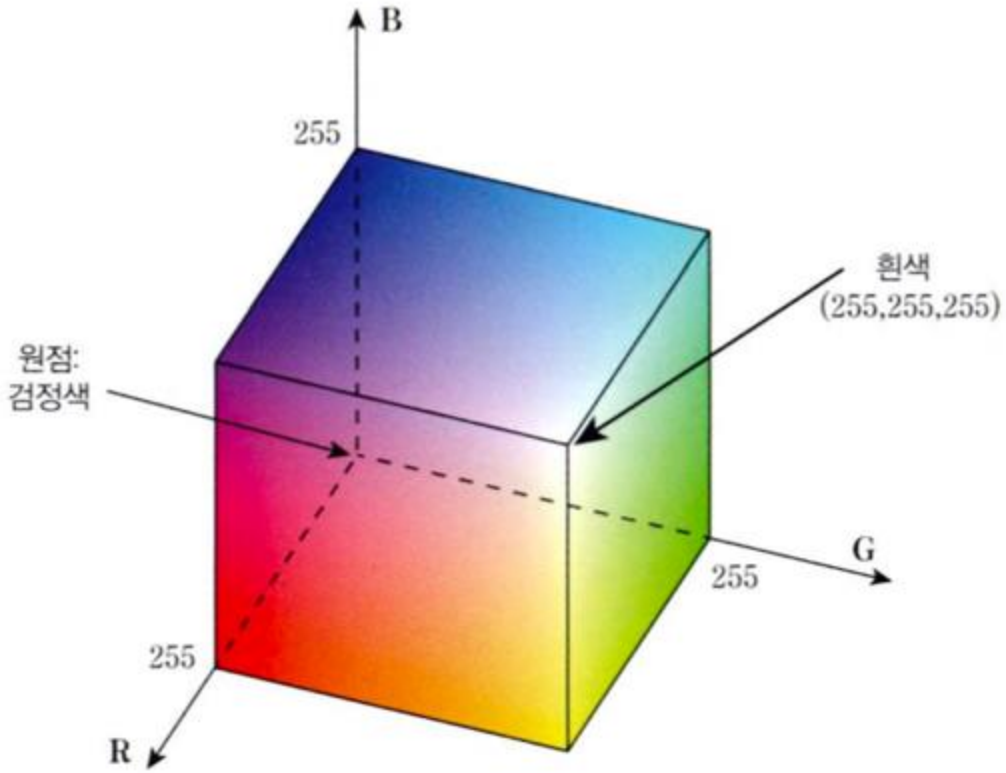
컬러 공간 변환

- RGB 컬러 공간
  - ✓ 빛을 이용해서 색을 만들려면 빨강 빛. 초록 빛, 파랑 빛이 반드시 필요한데 이것을 빛의 삼원색이라 부름
  - ✓ 원색이란 더이상 분해할 수 없는 최소한의 색을 의미하며 이 색들을 조합해서 다른 색을 표현할 수 있음
  - ✓ 이 원색의 조합은 그림과 같이 섞을 수록 밝아지기 때문에 가산 혼합(additive mixture)이라 함



컬러 공간 변환

- RGB 컬러 공간
  - ✓ 삼원색을 조합해서 다양한 색상을 표현할 수 있기 때문에 이 세 가지 색을 축으로 설정하여 컬러 공간을 만들 수 있는데 RGB 컬러 공간은 빨강색(Red), 녹색(Green), 파랑색(Blue)을 3개의 축을 직교하게 맞추어서 그림과 같이 입방체를 만들어 3차원 좌표계를 구성한 것



컬러 공간 변환

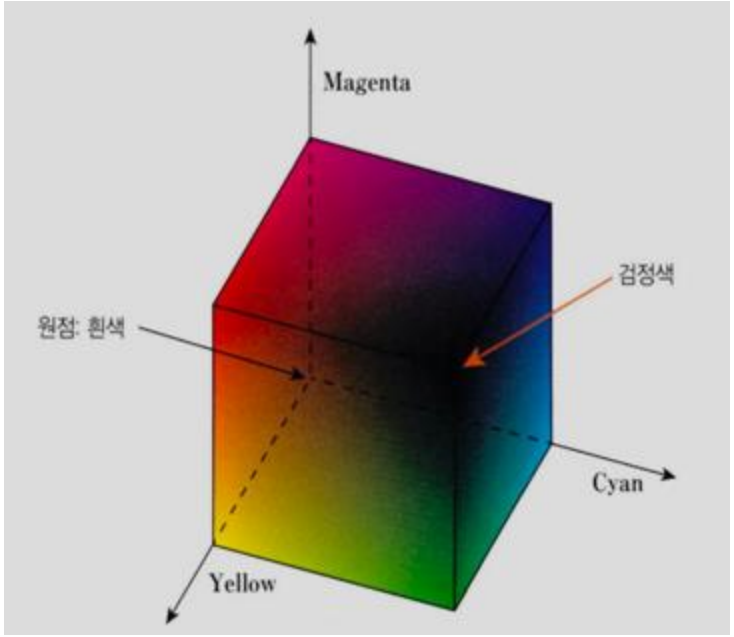
- RGB 컬러 공간
  - ✓ RGB 컬러 공간은 빛을 물리적으로 표현하고 만드는데 사용되는 기본 컬러 공간
  - ✓ 모니터에서 색을 표시하기 위해 이 컬러 공간을 사용
  - ✓ 영상 데이터를 모니터에 표시하기 위해서는 RGB 컬러 공간이 기본이 되는데 이런 이유에서 RGB 컬러 공간을 기본 컬러 공간이라 함
  - ✓ RGB 컬러 공간에서 3개 숫자로 표현된 값은 인간의 시각과 인지 체계로 정확한 색상을 규정하기가 쉽지 않으며 또한 이것은 각 채널의 상호 관계가 너무 크기 때문에 몇몇 영상처리 알고리즘들은 다른 컬러 공간을 사용
  - ✓ 히스토그램 평활화 와 같은 많은 영상처리 기술들은 영상의 명암도 요소만으로 처리가 가능

컬러 공간 변환

- ❖ CMY(K) 컬러 공간
  - ✓ 포토샵. 일러스트레이터, 페인트샵과 같은 그래픽 편집 툴을 사용한다면 작업한 내용들을 보기 위해서 모니터에 표시해야 하기 때문에 RGB 컬러 모델을 사용하는 것이 당연하지만 포토샵으로 작업한 내용을 인쇄소에서 출력할 때에는 빛이 아니라 물감의 색을 사용
  - ✓ 색은 섞으면 섞을수록 어두워지기 때문에 감산 혼합이라 하는데 인쇄소에서 빛의 원리가 적용된 RGB 컬러 공간을 그대로 사용하면 색상이 맞지 않아 잘못된 색상으로 인쇄되는 문제가 발생
  - ✓ CMY(K) 컬러 공간에서 필요한 것은 색의 삼원색
  - ✓ 색의 삼원색은 빛이 물체에서 반사되었을 때의 흡수되고 남은 색에 관한 것
  - ✓ 그림과 같이 색의 삼원색은 빛의 삼원색과 보색 관계에 있는 청록색(Cyan), 자홍색(Magenta), 노랑색(Yellow)를 의미
  - ✓ 이 원색들은 흰색 으로부터 감산되어 색이 만들어지기 때문에 감산 혼합(subtractive mixture)

컬러 공간 변환

- CMY(K) 컬러 공간
  - ✓ CMY 컬러 공간은 그림과 같이 색의 삼원색을 3개의 축으로 구성하여 입방체를 만들어 3차원 좌표계를 형성
  - ✓ RGB 컬러 공간과는 반대로 세 축이 교차하는 원점이 흰색이며 입방체의 대각선 반대쪽 끝이 검은색
  - ✓ 각 축은 그 색상의 밝기를 0∼255 사 이의 값으로 표현하며 세 축의 밝기 값이 교차하는 입방체 내부의 특정 좌표가 해당 색상이 됨



컬러 공간 변환

- CMY(K) 컬러 공간
  - ✓ CMY 컬러 공간과 RGB 컬러 공간은 보색 관계에 있기 때문에 두 컬러 공간은 쉽게 변환할 수 있음
  - ✓ 흰색에 대한 보수를 취하면 변환이 됨
  - ✓ 8비트 픽셀에서 흰색은 255이기 때문에 255에서 각 픽셀 값을 빼면 됨

    C=255 - R　　R=255 - C
    M=255 - G　　G=255 - M
    Y=255 – B　　B=255 – Y
  - ✓ 색의 삼원색을 모두 섞더라도 잉크에 포함되어 있는 불순물 등의 영향으로 완벽하게 검은색이 되지 않음
  - ✓ 순수한 검은색을 출력하기 위해서 CMY 컬러 모델에 검은색(blacK)을 추가하여 CMYK 컬러 공간으로 사용하는 것이 일반적
  - ✓ 순수한 검정색은 뛰어난 대비를 제공하며 검정 잉크가 컬러 잉크보다 비용이 적은 장점도 있음
  - ✓ CMY에서 CMYK 컬러 공간으로 변환하는 수식으로 blacK은 단순히 검정색이 아니라 다음과 같이 CMY 요소 중에서 최소값(min)을 의미

    blacK = min(Cyan, Magenta, Yellow)
    Cyan = Cyan - blacK
    Magenta = Magenta - blacK
    Yellow = Yellow - blacK

컬러 공간 변환

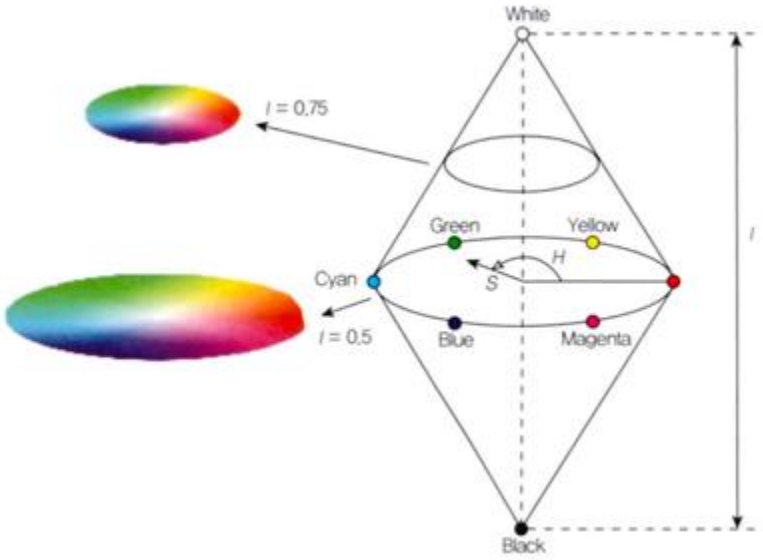
- ❖ CMY(K) 컬러 공간
  - ✓ BGR-CMY 컬러 변환

컬러 공간 변환

- CMY(K) 컬러 공간
  - ✓ BGR-CMYK 컬러 변환

컬러 공간 변환

- HSI 컬러 공간
  - ✓ 인간이 컬러 영상 정보를 인지하는 방법은 색상(Hue), 채도(Saturation), 명도(Intensity)라는 세 가지 지각 변수로 분류
  - ✓ 세 가지 속성을 그림과 같이 3차원 좌표에서 각각의 축으로 공간을 형성할 수 있음
  - ✓ 원뿔 모양의 공간 좌표계로 모형화한 것이 HSI 컬러 공간이며 세 가지 속성의 영문 첫 글자를 가져와서 이름을 붙인 것



컬러 공간 변환

- HSI 컬러 공간
  - ✓ 인간이 색상을 인식하는 3가지 요인인 색상. 채도, 명도를 컬러 공간으로 옮긴 것이기에 HSI 컬러 공간은 인간 시각 시스템 특성과 유사
  - ✓ 색상은 빛이 물체에서 반사되어 나온 색을 말하는 것으로 파장을 시각적으로 표현한 것
  - ✓ HSI 컬러 공간에서 색상은 원판의 0～360도까지 회전하며 표현
  - ✓ 0도가 빨간 색, 60도는 노란색, 120도는 녹색, 180도는 청록(Cyan), 240도는 파란색, 300도가 다홍 (Magenta)
  - ✓ 채도는 색의 순수한 정도를 나타내는데 순색(pure color)에 흰색의 혼합 비율에 따라서 0～100까지의 값을 가지는데 예를 들어 빨간색은 순색으로 채도가 높아서 100의 값을 갖게 되며 핑크색은 흰색이 포함된 정도에 따라서 그 값이 0에 가까워 짐
  - ✓ 컬러 공간에서는 채도는 원판의 반지름으로 표현되는데 원판의 중심이 0으로 가장 순도가 낮으며 가장 자리가 100으로 표현되고 채도가 가장 높은 순색
  - ✓ 명도는 빛의 세기라고 하며 색의 밝고 어두운 정도를 나타내는데 컬러 공간에서는 원뿔의 높이에 해당하며 가장 아래쪽 0이 검은색이며 가장 위쪽이 100으로 흰색을 표현하며 중간의 값들은 연한 회색에서부터 진한 회색까지의 스케일을 나타냄

컬러 공간 변환

- HSI 컬러 공간
  - ✓ 기본 컬러 공간인 RGB 컬러 공간에서 HSI 컬러 공간으로 변경하려면 다음과 같은 수식을 적용

$$\theta = \cos -1\left[\frac{((R-G)+(R-B))*0.5}{\sqrt{(R-G)^2+(R-B)\cdot(G-B)}}\right]$$

$$H = \begin{Bmatrix} \theta & , \text{ if } B \le G \\ 360-\theta, & otherwise \end{Bmatrix}$$

$$S = 1 - \frac{3\cdot \min(R,G,B)}{(R+G+B)}$$

$$I = \frac{1}{3}(R+G+B)$$

- ❑ R은 Red 채널, G는 Green 채널, B는 Blue 채널의 픽셀값
- ❑ H는 Hue 채널 픽셀값이며, S는 Saturation 채널, I는 Intensity 채널 픽셀값

컬러 공간 변환

- HSI 컬러 공간
  - ✓ OpenCV에서는 HSV와 HLS 컬러 공간에 대한 변환이 제공되며 HSI 컬러 공간 변환과 수식이 조금씩 다음
  - ✓ 아래 수식은 HSV 컬러 공간에 대한 변환 수식

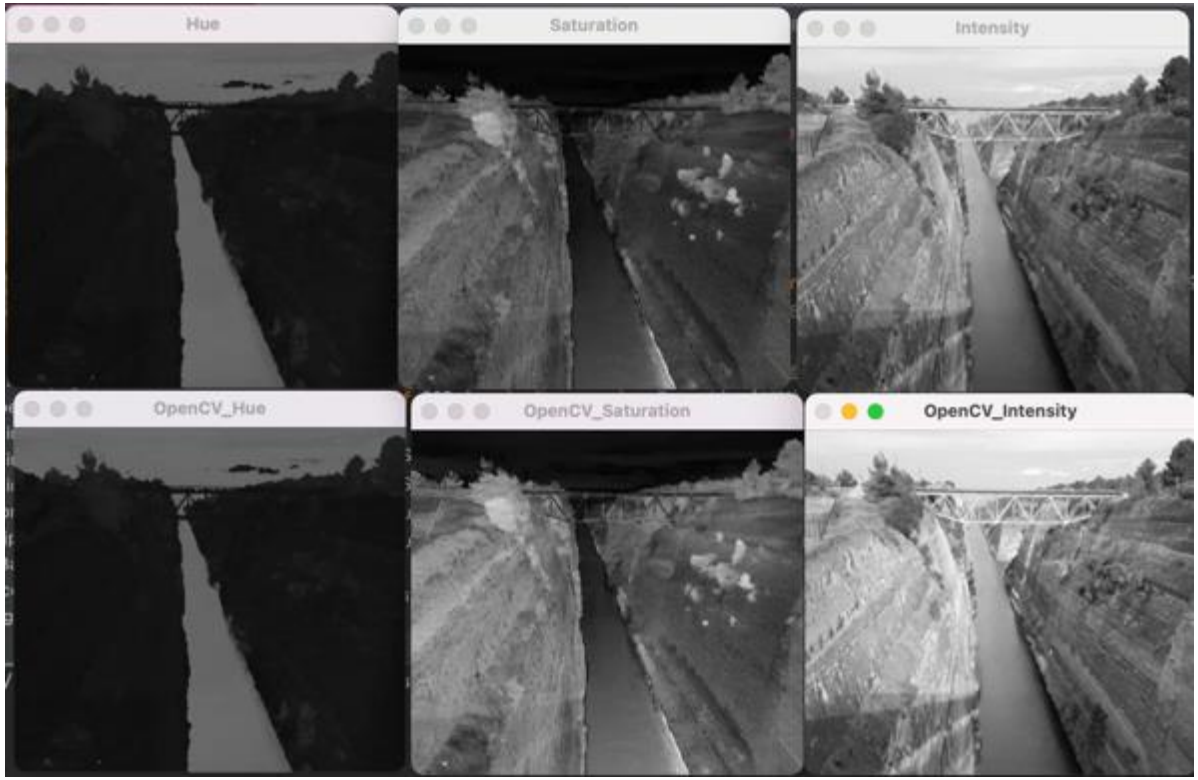
$$H = \begin{cases} \dfrac{(G-B)*60}{S}, & \text{if } V = R \\ \dfrac{(G-B)*60}{S} + 120, & \text{if } V = G \\ \dfrac{(G-B)*60}{S} + 240, & \text{if } V = B \end{cases}$$

$$S = \begin{cases} V - \dfrac{\min(R,G,B)}{V}, & \text{if } V \neq 0 \\ 0, & otherwise \end{cases}$$

$$V = \max(R,G,B)$$

컬러 공간 변환

- ❖ HSI 컬러 공간
  - ✓ BGR -> HSV 컬러 공간 변환



컬러 공간 변환

- ❖ 기타 컬러 공간
  - ✓ YCrCb 컬러 공간
    - ❑ 영상 시스템에서 사용되는 색 공간의 일종
    - ❑ Y는 휘도 성분이며 Cb와 Cr은 색차 성분
    - ❑ 인간의 시각은 밝기에는 민감하지만 색상에는 덜 민감한데 이러한 점을 이용해서 YCrCb 컬러 공간에서는 색차 신호인 Cr, Cb 성분을 Y 성분보다 상대적으로 낮은 해상도로 구성해서 인간의 시각에서 화질의 큰 저하 없이 영상 데이터의 용량을 감소할 수 있는데 휘도에 비해 색차 신호를 저해상도로 구성하는 방법은 간단하지만 효과적인 영상의 압축 방법으로 JPEG이나 MPEG에서 압축을 위한 기본 컬러 공간으로 YCbCr 컬러 공간이 이용됨
    - ❑ OpenCV의 cv2.cvtColor() 함수에서 RGB와 YCbCr를 서로 변환하는 수식

$$K = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

$$Cb = (R - Y) \cdot 0.564 + 128$$

$$Cr = (B - Y) \cdot 0.713 + 128$$

$$R = Y + 1.403 \cdot (Cr - 128)$$

$$G = X - 0.714 \cdot (Cr - 128) - 0.344(Cb - 128)$$

$$B = Y + 1.773 \cdot (Cb—128)$$

컬러 공간 변환

- 기타 컬러 공간
  - ✓ YUV 컬러 공간
    - TV 방송 규격에서 사용하는 컬러 표현 방식
    - PAL 방식의 아날로그 비디오를 위해 개발되었지만 디지털 비디오에서도 유럽의 비디오 표준으로 사용하고 있음
    - 변환 수식은 ITU-R BT.709 표준안에 따르는 것으로 HDTV에 적용되는 변환 수식

$$V = +0.2160 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B$$

$$U = -0.0999 \cdot R - 0.3360 \cdot G + 0.4360 \cdot B$$

$$V = +0.6150 \cdot R - 0.5586 \cdot G - 0.05639 \cdot B$$

$$R = Y + 1.28033 \cdot V$$

$$G = Y - 0.21482 \cdot U - 0.38059 \cdot V$$

$$B = Y + 2.12798 \cdot U$$

컬러 공간 변환

- ❖ 기타 컬러 공간
  - ✓ RGB 신호를 다른 컬러 공간으로 변환하는 수식은 디스플레이 장비나 기타 상황에 따라서 세부 비율이 조금씩 다르게 적용될 수 있음
  - ✓ 다른 컬러 공간으로는 스마트폰 카메라 등에서 주로 사용하는 YUV420, 비전 카메라에서 주로 사용하는 BayerRGB, 장치 독립 컬러 공간으로 알려진 La*b* 등 다양하게 존
  - ✓ OpenCV에서는 cv2.cvtColor() 함수에 옵션 상수(code)를 통해서 다양한 컬러 공간으로 변환할 수 있음

옵션 상수	값	옵션 상수	값	옵션 상수	값
COLOR_BGR2BGRA	0	COLOR_BGR2YCrCb	36	COLOR_HSV2BGR	54
COLOR_BGRA2BGR	1	COLOR_RGB2YCrCb	37	COLOR_HSV2RGB	55
COLOR_BGRA2RGB	2	COLOR_YCrCb2BGR	38	COLOR_LAB2BGR	56
COLOR_RGBA2BGR	3	COLOR_YCrCb2RGB	39	COLOR_LAB2RGB	57
COLOR_BGR2RGB ,	4	COLOR_BGR2HSV	40	COLOR_LUV2BGR	58
COLOR_BGRA2RGBA	5	COLOR_RGB2HSV	41	COLOR_LUV2RGB	59
COLOR_BGR2GRAY	6	COLOR_BGR2LAB	44	COLOR_HLS2BGR	60
COLOR_RGB2GRAY	7	COLOR_RGB2LAB	45	COLOR_HLS2RGB	61
COLOR_GRAY2BGR	8	COLOR_BayerBG2BGR	46	COLOR_BGR2YUV	82
COLOR_GRAY2BGRA	9	COLOR_BayerGB2BGR	47	COLOR_RGB2YUV	83
COLOR_BGRA2GRAY	10	COLOR_BayerRG2BGR	48	COLOR_YUV2BGR	84
COLOR_RGBA2GRAY	11	COLOR_BayerGR2BGR	49	COLOR_YUV2RGB	85
COLOR_BGR2XYZ	32	COLOR_BGR2LUV	50	COLOR_BayerBG2GRAY	86
COLOR_RGB2XYZ	33	COLOR_RGB2LUV	51	COLOR_BayerGB2GRAY	87
COLOR_XYZ2BGR	34	COLOR_BGR2HLS	52	COLOR_BayerRG2GRAY	88
COLOR_XYZ2RGB	35	COLOR_RGB2HLS	53	COLOR_BayerGR2GRAY	89

컬러 공간 변환

- ❖ 기타 컬러 공간
  - ✓ Hue 채널을 이용한 객체 검출