1.

|  |
| --- |
| 단일채널의 영상이다. |

2.

|  |
| --- |
| 화소의 밝기는 화소 값에 따라 변화하게 된다. 화소 값은 0~255의 값을 가지고 0에서부터 255로 갈수록 밝기가 밝아진다. 즉, 0은 검은색, 255은 흰색을 나타낸다. |

3.

|  |
| --- |
| 1. dst(y, x) = image1(y, x)\*0.5 + image2(y, x)\*0.5;  2. dst(y, x) = image1(y, x)\*alpha + image2(y, x)\*(1-alpha);  3. dst(y, x) = image1(y, x)\*alpha + image2(y, x)\*beta; |

4.

|  |
| --- |
| 영상에서 밝기를 변경하게 되면 영상의 모든 화소 값 같은 값만큼 증가하지만  명암 대비 변경에 경우에는 어두운 화소는 더 어둡게 되고 밝은 화소는 더 밝게 변한다. |

5.

|  |
| --- |
| 어떤 데이터가 얼마나 많은지를 나타내는 도수분포표를 그래프로 나타낸 것.  주로 세로축은 도수(화소 수), 가로축은 계급(화소 값)으로 나타낸다. |

6.

|  |
| --- |
| 1. 가장 높은 화소 값은 255, 가장 낮은 값은 0으로 값을 변경한다.  2. 중간의 화소 값들은 비율에 따라 값을 변경한다.  화소값= |

7.

|  |
| --- |
| 1. 히스토그램 계산  2. 히스토그램 빈도값에서 누적 빈도수를 계산  3. 누적 빈도수를 정규화한다.  4. 결과 화소값 = 정규화 누적합 \* 최대 화소값 |

8.

|  |
| --- |
| import numpy as np, cv2  img1=cv2.imread('C:/computervision/chap06/image/img1.jpg', cv2.IMREAD\_GRAYSCALE) img2=cv2.imread('C:/computervision/chap06/image/img2.jpg', cv2.IMREAD\_GRAYSCALE) if img1 is None or img2 is None : raise Exception("이미지읽기 오류") img1=img1[100:500, 100:500] img2=img2[100:500, 300:700]  alpha, beta = 0.7, 0.3 add\_img=cv2.addWeighted(img1, alpha, img2, beta, 0)  dst=np.zeros((img1.shape[0], img1.shape[1]\*3), np.uint8) dst[:400, :400]=img1[:400, :400] dst[:400, 400:800]=img2[:400, :400] dst[:400, 800:1200]=add\_img[:400, :400]  cv2.imshow('dst', dst) cv2.waitKey(0) |

9.

|  |
| --- |
| import numpy as np, cv2  img1=cv2.imread('C:/computervision/chap06/image/img1.jpg', cv2.IMREAD\_GRAYSCALE) img2=cv2.imread('C:/computervision/chap06/image/img2.jpg', cv2.IMREAD\_GRAYSCALE) if img1 is None or img2 is None : raise Exception("이미지읽기 오류") img1=img1[100:500, 100:500] img2=img2[100:500, 300:700] cv2.namedWindow('dst')  def image1(value):  pass def image2(value):  pass  alpha, beta = 50, 50 add\_img = cv2.addWeighted(img1, alpha/100, img2, beta/100, 0) dst=np.zeros((img1.shape[0], img1.shape[1]\*3), np.uint8) dst[:400, :400]=img1[:400, :400] dst[:400, 400:800]=img2[:400, :400]  cv2.createTrackbar('img1', 'dst', alpha, 100, image1) cv2.createTrackbar('img2', 'dst', beta, 100, image2)  while True:  if cv2.waitKey(30) >=0: break   alpha = cv2.getTrackbarPos('img1', 'dst')  beta = cv2.getTrackbarPos('img2', 'dst')  add\_img = cv2.addWeighted(img1, alpha/100, img2, beta/100, 0)  dst[:400, 800:1200]=add\_img[:400, :400]  cv2.imshow('dst', dst) |

10.

|  |
| --- |
| import numpy as np, cv2  image1 = np.zeros((50, 512), np.float32) image2 = np.zeros((50, 512), np.float32) rows, cols = image1.shape[:2]  for i in range(rows):  for j in range(cols):  image1.itemset((i, j), 1-j/512)  print(j/512,"\n")  cv2.imshow('image1', image1) cv2.waitKey(0) |

11.

|  |
| --- |
|  |

12.

|  |
| --- |
| import numpy as np, cv2  image = cv2.imread("c:/computervision/chap06/image/img1.jpg", 0) if image is None : raise Exception("영상파일 읽기 오류")  projection\_h = np.zeros((image.shape[0], 256), np.uint8) list\_h = cv2.reduce(image, dim=1, rtype=cv2.REDUCE\_AVG) list\_h = list\_h.flatten() print(list\_h) for i in range(image.shape[0]):  cv2.line(projection\_h, (0, i), (list\_h[i], i), 255, 1)  projection\_w = np.zeros((256, image.shape[1]), np.uint8) list\_w = cv2.reduce(image, dim=0, rtype=cv2.REDUCE\_AVG) list\_w = list\_w.flatten() for i in range(image.shape[1]):  cv2.line(projection\_w, (i, 0), (i, list\_w[i]), 255, 1) cv2.flip(projection\_w, 0, projection\_w)  cv2.imshow('1', projection\_h) cv2.imshow('2', projection\_w) cv2.imshow('3', image) cv2.waitKey() |

13.

|  |
| --- |
| import numpy as np, cv2  image = cv2.imread("c:/computervision/chap06/image/img1.jpg", 1) if image is None : raise Exception("영상파일 읽기 오류")  Y = np.zeros((image.shape[0],image.shape[1]), np.float32) Cb = np.zeros((image.shape[0],image.shape[1]), np.float32) Cr = np.zeros((image.shape[0],image.shape[1]), np.float32)  b, g, r = cv2.split(image) Y = 0.299\*r + 0.587\*g + 0.114\*b Cb = (r-Y)\*0.564 + 128 Cr = (b-Y)\*0.713 + 128  Y = np.round(Y) Y = Y.astype('uint8') Cb = np.round(Cb) Cb = Cb.astype('uint8') Cr = np.round(Cr) Cr = Cr.astype('uint8')  zero = np.zeros((image.shape[0],image.shape[1]), np.uint8) YCbCr = cv2.merge([Y, Cr, Cb]) YCC = cv2.split(YCbCr)  cv2.imshow('YCbCr', YCbCr) cv2.imshow('Y', YCC[0]) cv2.imshow('Cr', YCC[1]) cv2.imshow('Cb', YCC[2]) cv2.waitKey(0) |

14.

|  |
| --- |
| import numpy as np, cv2  image = cv2.imread("c:/computervision/chap06/image/img1.jpg", 1) if image is None : raise Exception("영상파일 읽기 오류")  HSV = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV) H, S, V = cv2.split(HSV)  dst = np.zeros((180,256, 3), np.uint8) dst\_H, dst\_S, dst\_V = cv2.split(dst)  for y in range(HSV.shape[0]):  for x in range(HSV.shape[1]):  dst\_V[H[y,x],S[y, x]]+=1  for y in range(180):  for x in range(256):  dst\_H[y, x] = y  dst\_S[y, x] = x  dst = cv2.merge((dst\_H, dst\_S, dst\_V)) dst = cv2.cvtColor(dst, cv2.COLOR\_HSV2BGR)  cv2.imshow('1', dst) cv2.waitKey(0) |