import cv2

import numpy as np

print ("Max of 255:"+ str(cv2.add(np.uint8([200]), np.uint8 ([100]))))

print ('Min of 0:' + str(cv2.subtract(np.uint8([50]), np.uint8 ([100]))))

print ('Around:' + str(np.uint8([200]) + np.uint8([100])))

print ('Around:' + str(np.uint8([50]) - np.uint8([100])))

Max of 255:[[255]]

Min of 0:[[0]]

Around:[44]

Around:[206]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Image Arithmetic and Brigthness

import cv2

import numpy as np

image=cv2.imread('c:\\users\\egek\\ders\\images\\rot.jpg')

cv2.imshow("Original", image)

M = np.ones(image.shape, dtype = "uint8") \* 100

added = cv2.add(image, M) #parlaklık arttırma

cv2.imshow("Added", added)

M = np.ones(image.shape, dtype = "uint8") \* 50

subtracted = cv2.subtract(image, M) #parlaklık azaltma

cv2.imshow("Subtracted", subtracted)

cv2.waitKey(0)

cv2.destroyAllWindows()

#Image Arithmetic

import cv2

import numpy as np

image=cv2.imread('c:\\users\\egek\\ders\\images\\rot.jpg')

cv2.imshow("Original", image)

new\_image = np.zeros(image.shape, image.dtype)

alpha=0.5 #contrast azaltma

beta=0 #parlaklık

for y in range(image.shape[0]):

for x in range(image.shape[1]):

new\_image[y,x] = np.clip(alpha\*image[y,x] + beta, 0, 255)

new\_image2 = np.zeros(image.shape, image.dtype)

cv2.imshow("Contrast", new\_image)

alpha=1.25 #contrast arttırma

beta=0 #parlaklık

for y in range(image.shape[0]):

for x in range(image.shape[1]):

new\_image2[y,x] = np.clip(alpha\*image[y,x] + beta, 0, 255)

cv2.imshow("Contrast2", new\_image2)

cv2.waitKey(0)

cv2.destroyAllWindows()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Bitwise Operators

import numpy as np

import cv2

rect = np.zeros((300, 300), dtype = "uint8")

cv2.rectangle(rect, (25, 25), (275, 275), (255,255,255), -1)

cv2.imshow("Rectangle", rect)

cir = np.zeros((300, 300), dtype = "uint8")

cv2.circle(cir, (150, 150), 150, (255,255,255), -1)

cv2.imshow("Circle", cir)

cv2.waitKey(0)

cv2.destroyAllWindows()

#Bitwise Operators

import numpy as np

import cv2

rect = np.zeros((300, 300), dtype = "uint8")

cv2.rectangle(rect, (25, 25), (275, 275), 255, -1)

cv2.imshow("Rectangle", rect)

cir = np.zeros((300, 300), dtype = "uint8")

cv2.circle(cir, (150, 150), 150, 255, -1)

cv2.imshow("Circle", cir)

bitwiseAnd = cv2.bitwise\_and(rect, cir)

cv2.imshow("AND", bitwiseAnd)

cv2.waitKey(0)

bitwiseOr = cv2.bitwise\_or(rect, cir)

cv2.imshow("OR", bitwiseOr)

cv2.waitKey(0)

bitwiseXor = cv2.bitwise\_xor(rect, cir)

cv2.imshow("XOR", bitwiseXor)

cv2.waitKey(0)

bitwiseNot = cv2.bitwise\_not(cir)

cv2.imshow("NOT", bitwiseNot)

cv2.waitKey(0)

cv2.destroyAllWindows()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Masking

import cv2

import numpy as np

image=cv2.imread('c:\\users\\egek\\ders\\images\\rot.jpg')

cv2.imshow("Original", image)

mask = np.zeros(image.shape, dtype = "uint8")

(cX, cY) = (image.shape[1] // 2, image.shape[0] // 2)

#cv2.rectangle(mask, (cX - 75, cY - 75), (cX + 75 , cY + 75), (255,255,255), -1)

cv2.circle(mask, (cX,cY),100, (255,255,255), -1)

cv2.imshow("Mask", mask)

masked = cv2.bitwise\_and(image,mask)

cv2.imshow("Mask Applied to Image", masked)

cv2.waitKey(0)

cv2.destroyAllWindows()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Splitting and Merging Channels

import cv2

import numpy as np

image=cv2.imread('c:\\users\\egek\\ders\\images\\rot.jpg')

(B, G, R) = cv2.split(image)

cv2.imshow("Original", image)

zeros = np.zeros((image.shape[0],image.shape[1],1), dtype = "uint8")

cv2.imshow("Red", cv2.merge([zeros, zeros, R]))

cv2.imshow("Green", cv2.merge([zeros, G, zeros]))

cv2.imshow("Blue", cv2.merge([B, zeros, zeros]))

cv2.waitKey(0)

merged = cv2.merge([B, G, R])

cv2.imshow("Merged", merged)

cv2.waitKey(0)

cv2.destroyAllWindows()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Color Spaces

import cv2

import numpy as np

image=cv2.imread('c:\\users\\egek\\ders\\images\\rot.jpg')

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

cv2.imshow("Gray", gray)

hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

cv2.imshow("HSV", hsv)

lab = cv2.cvtColor(image, cv2.COLOR\_BGR2LAB)

cv2.imshow("L\*a\*b\*", lab)

cv2.waitKey(0)

cv2.destroyAllWindows()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#PYRAMID

import cv2

image = cv2.imread('c:\\users\\egek\\ders\\images\\rot.jpg')

smaller = cv2.pyrDown(image)

larger = cv2.pyrUp(smaller)

cv2.imshow('Original', image )

cv2.imshow('Smaller ', smaller )

cv2.imshow('Larger ', larger )

cv2.waitKey(0)

cv2.destroyAllWindows()