Assignment5

Color Histogram Equalization

2017113547 이정근

#Color Histogram Equalization.py

#2017113547 이정근

```
import cv2
import numpy as np
NUM_INTENSITY = 256


def HistogramEqualize(img):
    height, width = img.shape
    num_pixels = height*width

    # making histogram
    histogram = np.zeros((NUM_INTENSITY,))
    for y in range(height):
        for x in range(width):
            histogram[img[y, x]] = histogram[img[y, x]] + 1

    # normalizing histogram
    normalized_histogram = np.divide(histogram, num_pixels)
```

```
    # making cdf
    cdf = np.zeros((NUM_INTENSITY,))
    cdf[0] = normalized_histogram[0]
    for k in range(1, NUM_INTENSITY, 1):
        cdf[k] = cdf[k-1] + normalized_histogram[k]

    # finding output gray level
    output_gray_level = np.multiply(NUM_INTENSITY-1, cdf)
    output_gray_level = np.round(output_gray_level)

    # convert image to equalized image
    result = np.zeros((height, width), np.uint8)   # result image
    for y in range(height):
        for x in range(width):
            result[y, x] = output_gray_level[img[y, x]]

    return result


img = cv2.imread('dgu_night_color.png', cv2.IMREAD_COLOR)   # img2numpy
imgYCC = cv2.cvtColor(img, cv2.COLOR_BGR2YCR_CB)
imgYCC_out = np.zeros(imgYCC.shape, np.uint8)
imgYCC_out[:,:,0] = HistogramEqualize(imgYCC[:,:,0])

row, col = imgYCC_out[:,:,0].shape

# divide by zero error preventing
for i in range(row):
    for j in range(col):
        if imgYCC[i, j, 0] == 0:
            imgYCC[i, j, 0] = 1

s = 0.095
img_out = np.zeros(img.shape, np.uint8)

img_out[:,:,0] = np.multiply(imgYCC_out[:,:,0],np.divide(img[:,:,0],imgYCC[:,:,0])**s)
img_out[:,:,1] = np.multiply(imgYCC_out[:,:,0],np.divide(img[:,:,1],imgYCC[:,:,0])**s)
img_out[:,:,2] = np.multiply(imgYCC_out[:,:,0],np.divide(img[:,:,2],imgYCC[:,:,0])**s)
```

```
cv2.imshow("Color Histogram Equalized Image", img_out)
cv2.waitKey()
```