

به نام خدا

درس: بینایی کامپیوتر

عنوان تمرین: متعادل سازی عکس

نام و نام خانوادگی: امیرمسعود شاکر

شماره دانشجویی: 97243081

در این تمرین، خواسته شده الگوریتم متعادل سازی عکس پیاده سازی شده و نتیجه اجرای آن روی یک عکس سیاه و سفید با نتیجه استفاده از تابع آماده کتابخانه PIL مقایسه شود.

کد این تمرین به صورت زیر است:

```
from PIL import Image, ImageOps
import numpy as np

### My implementation of Histogram Equalization ###

img = Image.open('./flowers/flowers.webp').convert('L') # read a gray scale image
arrayed_img = np.asarray(img) # convert the image to a numpy array
# print(arrayed_img)
# print(arrayed_img.shape)
M = arrayed_img.shape[0]
N = arrayed_img.shape[1]

flattened_img = arrayed_img.flatten()
# print(flattened_img)

gray_levels_arr = np.unique(flattened_img) # find the unique values in the whole array
# print(gray_levels_arr)
G = gray_levels_arr.max() # maximum gray level
# print(G)

num_of_pixels = len(flattened_img)
pixel_occurrences = np.zeros(G+1)
cumulative_occurrences = np.zeros(G+1)

i = 0
for gray_level in gray_levels_arr:
```

```

    pixel_occurrences[gray_level] = np.count_nonzero(flattened_img == gray_level)
    i += 1

# print(pixel_occurrences)

cumulative_occurrences[0] = pixel_occurrences[0]

for i in range(1, G+1):
    cumulative_occurrences[i] = pixel_occurrences[i] + cumulative_occurrences[i-1]

cumulative_occurrences = np.round(cumulative_occurrences * G /
    cumulative_occurrences[len(cumulative_occurrences)-1])
# print(cumulative_occurrences)

i = 0
for pixel in flattened_img:
    for gray_level in gray_levels_arr:
        if pixel == gray_level:
            flattened_img[i] = cumulative_occurrences[gray_level]
            i += 1
            break

output_img = flattened_img.reshape((M, N))
# print(output_img)

equalized_img = Image.fromarray(output_img)
equalized_img.save('./flowers/equalized_by_masoud.webp')

### compare my result with PIL's ###
PIL_equalized_img = ImageOps.equalize(img) # equalized image using PIL package
equalized_img.save('./flowers/equalized_by_PIL.webp')

equalized_img.show()

```

در ادامه نتیجه اجرای کد روی سه عکس مختلف را با هم میبینیم (به ترتیب عکس اصلی، نتیجه اجرای پیاده سازی انجام شده و نتیجه اجرای تابع آماده کتابخانه PIL):







