تمرین پردازش تصاویر دیجیتالی

تمرین شماره ۴

تهیمنه توکلی

۹۹۱۲۷۶۲۲۶۷

موعد تحویل: ۰۳/۰۲/۱۷

تاریخ تحویل: ۰۳/۰۲/۱۵

شرح تکنیکی تمرین و نتایج

۱. **برنامه ای به زبان مطلب (پایتون یا هر زبان دیگری) بنویسید که یک تصویر سطح خاکستری را از ورودی دریافت کرده و عملیات زیر را روی آن انجام می دهد:**

الف- محاسبه بردار هیستوگرام و ترسیم نمودار آن

ب- اعمال روش همسان سازی هیستوگرام و محاسبه تابع همسان ساز

ج- اعمال تابع همسان ساز به تصویر ورودی و بدست آوردن تصویر جدید و نمایش آن.

تصویر سطح خاکستری plants.jpg از ورودی دریافت میشود.

الف) هیستوگرام احتمال وقوع سطوح خاکستری مختلف را نشان میدهد. هیستوگرام یک تصویر دیجیتالی با سطوح خاکستری در محدوده [L-1, 0] یک تابع گسسته است به طوری که:

$$h(r_k) = n_k$$

r_k نشان دهنده ی kامین سطح خاکستری و n_k تعداد پیکسل های تصویر که دارای سطح خاکستری r_k هستند. هیستوگرام نرمالیزه شده با تقسیم هر مقدار در هیستوگرام بر تعداد کل پیکسل های تصویر که با n نشان داده می شود بدست می آید. که حاوی اصلاعات آماری مفیدی در مورد تصویر می باشد.

محاسبه بردار هیستوگرام در تابع calculate.histogram() پیاده‌سازی شده است. (همچنین برای محاسبه بردار هیستوگرام میتوان از تابع آماده cv2.calcHist() استفاده کرد).

ترسیم نمودار هیستوگرام در plot_histogram() انجام میشود. نمودار هیستوگرام و هیستوگرام نرمالایز شده در مسیر ./histogrm.png/hist_۱ قرار دارد.ب

ب) تابع histogram_equalization() ابتدا هیستوگرام تصویر را محاسبه می‌کند و سپس تابع توزیع تجمعی (CDF) را برای تعیین تابع همسان ساز برای هر شدت پیکسل محاسبه می‌کند. (همچنین میتوان برای اعمال تابع همسان ساز از تابع آماده cv2.equalizeHist() استفاده کرد).

ج) تابع histogram_equalization() در نهایت، تابع تبدیل محاسبه شده را به تصویر اصلی اعمال می کند تا تصویر جدید را به دست آورد. تصویر جدید در مسیر ./eq_image_۱.jpg قرار دارد.

۲. **مطلوبست اعمال روش همسان سازی هیستوگرام به تصویر زیر. نمودار هیستوگرام تصویر را قبل و بعد از اعمال روش همسان سازی هیستوگرام ترسیم کنید. چه نتیجه ای می گیرید. در مورد هر نمودار توضیحاتی را ارائه کنید.**

تصویر اصلی در مسیر ./org_image.jpg قرار دارد. برای همسان سازی هیستوگرام از تابع histogram_equalization() استفاده شده. تصویر جدید در مسیر ./eq_image_۲.jpg قرار دارد.
ترسیم نمودار هیستوگرام در تابع plot_histogram() انجام میشود. نمودار هیستوگرام و هیستوگرام نرمالایز شده تصویر قبل و بعد از اعمال روش همسان سازی هیستوگرام در مسیر ./hist_۲/ قرار دارد.
نمودارهای هیستوگرام نمایش گرافیکی توزیع شدت پیکسل در یک تصویر هستند. آنها نشان می دهند که چند پیکسل در تصویر دارای چه مقدار شدت هستند.

قبل از یکسان سازی هیستوگرام، در هیستوگرام اصلی شدت پیکسل‌ها به‌طور نابرابر توزیع شده اند که می‌تواند منجر به نواحی بسیار تاریک یا خیلی روشن شود.

هیستوگرام به سمت مقادیر با شدت زیاد منحرف شده است که نشان می دهد قسمت های خاصی از تصویر بیش از حد نوردهی شده اند.

پس از یکسان سازی هیستوگرام، یکسان سازی هیستوگرام شدت پیکسل ها را دوباره توزیع می کند تا هیستوگرام متعادل تری ایجاد کند.

هیستوگرام یکسان شده یکنواخت تر است و شدت پیکسل ها را در کل محدوده پخش می کند. این فرآیند می تواند کنتراست و ظاهر کلی تصویر را بهبود بخشد و از نظر بصری جذاب تر شود.

به طور خلاصه، قبل از یکسان سازی هیستوگرام، هیستوگرام ممکن است کج یا ناهموار باشد، در حالی که پس از یکسان سازی هیستوگرام، هیستوگرام یکنواخت تر و متعادل تر می شود که منجر به بهبود کیفیت تصویر می شود.

پیوست کدهای پیاده سازی شده

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def load_image(image_path):
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    return image

def display_image(image):
    cv2.imshow("Image", image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
    return

def calculate_histogram(image, range=256):
    histogram = [0] * range
    for row in image:
        for pixel in row:
            histogram[pixel] += 1
    return histogram

def plot_histogram(image, path, range=256):
    # histogram = cv2.calcHist([image], [0], None, [range], [0, range])
    histogram = calculate_histogram(image, range)
    plt.plot(histogram, color="black")
    plt.xlabel("Pixel Intensity")
    plt.ylabel("Frequency")
```

```
    plt.title("Histogram")
    plt.savefig(f"{path}/histogram.png")
    plt.close()

    normalized_histogram = histogram / np.sum(histogram)
    plt.plot(normalized_histogram, color="black")
    plt.xlabel("Pixel Intensity")
    plt.ylabel("Normalized Frequency")
    plt.title("Normalized Histogram")
    plt.savefig(f"{path}/normalized_histogram.png")
    plt.close()

def histogram_equalization(image, r=256):
    histogram = calculate_histogram(image, range=r)

    # Compute cumulative distribution function (CDF)
    cdf = [sum(histogram[: i + 1]) for i in range(len(histogram))]

    # Compute histogram equalization transformation
    total_pixels = len(image) * len(image[0])
    equalized_image = [[0] * len(image[0]) for _ in range(len(image))]
    for i in range(len(image)):
        for j in range(len(image[0])):
            intensity = image[i][j]
            equalized_intensity = int(round((cdf[intensity] / total_pixels) * 255))
            equalized_image[i][j] = equalized_intensity

    equalized_image = np.array(equalized_image, dtype=np.uint8)
    return equalized_image
```

```
if __name__ == "__main__":
    # PART ONE
    image_path = "./plants.jpg"
    image = load_image(image_path)

    plot_histogram(image, path="./hist_1")

    # equalized_image = cv2.equalizeHist(image)
    equalized_image = histogram_equalization(image)
    cv2.imwrite("eq_image_1.jpg", equalized_image)

    # PART TWO
    array = np.array(
        [
            [1, 2, 2, 6, 5, 5, 6, 7, 3, 7],
            [1, 5, 6, 1, 2, 2, 2, 2, 7, 7],
            [2, 1, 6, 4, 2, 3, 4, 2, 4, 3],
            [3, 5, 3, 4, 4, 6, 6, 6, 6, 6],
            [4, 6, 7, 3, 4, 6, 5, 4, 5, 6],
            [6, 3, 2, 1, 7, 7, 4, 5, 6, 7],
            [2, 1, 6, 4, 2, 3, 4, 2, 4, 3],
            [1, 2, 2, 6, 5, 5, 6, 7, 3, 7],
            [4, 6, 7, 3, 4, 6, 5, 4, 5, 6],
            [6, 3, 2, 1, 7, 7, 4, 5, 6, 7],
        ]
    )
    image2 = np.uint8(array)
    cv2.imwrite("org_image.jpg", image2)
```

```
# equalized_image = cv2.equalizeHist(image2)
equalized_image2 = histogram_equalization(image2, r=8)
cv2.imwrite("eq_image_2.jpg", equalized_image2)

plot_histogram(image2, path="./hist_2/before", range=8)
plot_histogram(equalized_image2, path="./hist_2/after")
```