



تمرین پردازش تصاویر دیجیتالی

تمرین شماره ۳

تهیمنه توکلی

۹۹۱۲۷۶۲۲۶۷

موعد تحویل: ۰۳/۰۲/۱۴

تاریخ تحویل: ۰۳/۰۲/۱۲

شرح تکنیکی تمرین

۱. برنامه ای به زبان مطلب (پایتون یا هر زبان دیگری) بنویسید که تصویر **rice.png** را از ورودی گرفته و تابع نمایی بازای گاماهای با مقادیر زیر را به آن اعمال می کند و تصاویر خروجی را نمایش می دهد. سپس نتایج را تحلیل کنید.

گاما: ۰.۱, ۰.۲, ۰.۸, ۱.۵, ۳, ۵

در این قسمت برای اعمال تابع نمایی گاما هر پیکسل از تصویر ورودی به ۲۵۵ تقسیم شده و سپس به توان گاما میرسد. سپس در ۲۵۵ ضرب میشود تا پیکسل تصویر خروجی بدست آید. این تابع در `gamma_correction()` پیاده سازی شده است.

تصاویر خروجی در مسیر `results/rice_gamma_x.png/` قرار دارد. X نشان دهنده مقدار گاما اعمال شده به تصویر است.

۲. برای بهبود کیفیت تصویر زیر از تابع نمایی با چه گامایی استفاده می کنید. برای درستی پاسخ خود دلیل بیاورید. آیا راه حل بهتری سراغ دارید؟ توضیح دهید.

راه حل اول: با مشاهده تصویر، نوع گاما و روش آزمون و خطا مقدار مناسب آن را بیابیم.

مقادیر گاما ۰.۱, ۰.۲, ۰.۸, ۱.۵, ۳ و ۵ به تصویر اعمال شده است. تصاویر خروجی در مسیر `results/plants_gamma_x.png` قرار دارد. X نشان دهنده مقدار گاما اعمال شده به تصویر است.

راه حل دوم: انتخاب نوع گاما بر اساس میانگین تصویر و انتخاب مقداری که میانگین تصویر را ۱۲۸ کند.

میانگین تصویر اصلی ۶۹.۷۶۵ است. مقادیر مختلف گاما (۰.۱, ۰.۲, ۰.۳, ۰.۴, ۰.۵, ۰.۶) به تصویر اصلی اعمال شده و سپس میانگین تصویر مجدد محاسبه شده است:

| گاما | میانگین تصویر |
|------|---------------|
| 0.1 | 209.33 |
| 0.2 | 175.04 |
| 0.3 | 148.77 |
| 0.4 | 128.41 |
| 0.5 | 112.38 |
| 0.6 | 99.84 |

بنابراین بهترین مقدار گاما تقریباً ۰.۴ است.

نتایج

۱. برنامه ای به زبان مطلب (پایتون یا هر زبان دیگری) بنویسید که تصویر **rice.png** را از ورودی گرفته و تابع نمایی بازای گاماهای با مقادیر زیر را به آن اعمال می کند و تصاویر خروجی را نمایش می دهد. سپس نتایج را تحلیل کنید.

گاما: ۰.۱, ۰.۲, ۰.۸, ۱.۵, ۳, ۵

برای تحلیل نتایج میتوانیم تصاویر خروجی را بررسی کنیم و تصویری که به نظر بیشترین وضوح را دارد به عنوان گاما مناسب انتخاب کنیم. با توجه به نتایج گاما ۱.۵ وضوح بهتری دارد.

همچنین میتوانیم میانگین تصاویر را در نظر بگیریم. مقداری که میانگین تصویر را ۱۲۸ کند؛ مقدار گاما بهتری است.

| گاما | میانگین تصویر |
|------|---------------|
| 0.1 | 241.18 |
| 0.2 | 228.62 |
| 0.8 | 165.80 |
| 1.5 | 114.07 |
| 3 | 51.38 |
| 5 | 17.90 |

از بین این مقادیر گاما ۱.۵ نزدیکترین میانگین به ۱۲۸ را دارد. بنابراین بهترین گاما است.

با امتحان کردن مقادیر دیگر گاما میتوان نتیجه گرفت که ۱.۳ بهترین گاما برای اعمال به این تصویر است زیرا میانگین تصویر پس از اعمال تابع نمایی با گاما ۱.۳ برابر ۱۲۷.۰۱ میشود که به ۱۲۸ نزدیکتر است.

همچنین میتوانیم نمودار هیستوگرام تصاویر را رسم کنیم. هر تصویر که نمودار هیستوگرام هموارتری داشته باشد؛ بازنمایی بهتری داشته و آن مقدار گاما مناسب تر است. تصاویر خروجی در مسیر `hist/rice_gamma_x.png` قرار دارد. X نشان دهنده مقدار گاما اعمال شده به تصویر است.

البته از مشکلات اساسی تبدیل نمایی این است که برای تصاویر رنگ و رو رفته و تصاویری که کنتراست پایینی دارند خوب کار نمی کنند. به همین دلیل با توجه به اینکه تصویر اصلی رنگ و رو رفته است؛ نمیتوانیم با تبدیل نمایی گاما به بازنمایی خوبی از تصویر برسیم.

(علل پایین بودن کنتراست تصویر: نورپردازی ضعیف - تنظیم نادرست روزنه لنز (فوکوس) - تنظیم نادرست سرعت شاتر (میزان متوسط انرژی ورودی) - محدوده کوچک میزان دقت حسگر)

۲. برای بهبود کیفیت تصویر زیر از تابع نمایی با چه گامایی استفاده می کنید. برای درستی پاسخ خود دلیل بیاورید.
آیا راه حل بهتری سراغ دارید؟ توضیح دهید.

راه حل اول: با مشاهده تصویر، نوع گاما و روش آزمون و خطا مقدار مناسب آن را بیابیم. مقادیر گاما ۰,۱، ۰,۲، ۰,۳، ۰,۴، ۰,۵ و ۰,۸ به تصویر اعمال شده است. با توجه به تصاویر خروجی بهترین گاما ۰,۵ است.

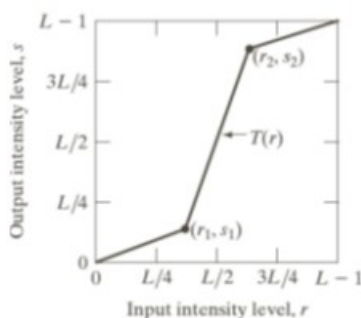
راه حل دوم: انتخاب نوع گاما بر اساس میانگین تصویر و انتخاب مقداری که میانگین تصویر را ۱۲۸ کند. میانگین تصویر اصلی ۶۹.۷۶۵ است. از بین مقادیر گاما اعمال شده مقدار ۰,۴ نزدیکترین میانگین به ۱۲۸ را دارد. بنابراین بهترین گاما است.

همچنین میتوانیم نمودار هیستوگرام تصاویر را رسم کنیم. هر تصویر که نمودار هیستوگرام هموارتری داشته باشد؛ بازنمایی بهتری داشته و آن مقدار گاما مناسب تر است. تصاویر خروجی در مسیر /hist/plants_gamma_x.png قرار دارد. X نشان دهنده مقدار گاما اعمال شده به تصویر است.

با توجه به اینکه مولفه های آماری تصویر مورد نظر مانند میانگین و واریانس در اکثر نقاط تصویر مقادیر متفاوتی دارند؛ تصویر غیر ایستادن (non stationary) است و روش تبدیل نمایی برای تصاویر غیرایستادن خوب کار نمی کند. به همین دلیل نمیتوانیم با تبدیل نمایی گاما به بازنمایی خوبی از تصویر برسیم.

راه حل ها:

الف- استفاده از توابع قطعه خطی: مثال برای تصاویر رنگ و رو رفته استفاده از تابع زیر



ب- استفاده از یک table up-look: برای تبدیل هر پیکسل تصویر مبدا به مقدار ایده آل در تصویر مقصد از این جدول استفاده می شود.

```
import cv2

import numpy as np

from matplotlib import 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 gamma_correction(image, gamma):

    # corrected_image = np.power(image / 255.0, 1 / gamma)

    corrected_image = np.power(image / 255.0, gamma)

    corrected_image = np.uint8(corrected_image * 255)

    return corrected_image

def plot_histogram(image, path, range=256):

    histogram = cv2.calcHist([image], [0], None, [range], [0, range])

    plt.plot(histogram, color="black")

    plt.xlabel("Pixel Intensity")

    plt.ylabel("Frequency")

    plt.title("Histogram")

    plt.savefig(path)
```

```

plt.close()

if __name__ == "__main__":
    # PART ONE

    image_path = "./rice.png"
    image = load_image(image_path)
    gamma = [0.1, 0.2, 0.8, 1.3, 1.5, 3, 5]

    for g in gamma:
        result = gamma_correction(image, g)
        cv2.imwrite(f"./results/rice_gamma_{g}.png", result)
        print(g, np.mean(result))

    # 0.1: 241.18221184248347
    # 0.2: 228.6210074734119
    # 0.8: 165.80338603046852
    # 1.3: 127.01247484909457
    # 1.5: 114.07638401839608
    # 3: 51.38579764300086
    # 5: 17.902375682667433

    # PART TWO

    image_path = "./plants.jpg"
    image = load_image(image_path)

    for g in gamma:
        result = gamma_correction(image, g)
        cv2.imwrite(f"./results/plants_gamma_{g}.png", result)

    gamma = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6]

```

```

for g in gamma:

    result = gamma_correction(image, g)

    cv2.imwrite(f"./results/plants_gamma_{g}.png", result)

    print(g, np.mean(result))

# 0.1: 209.3303697630742
# 0.2: 175.04046843544427
# 0.3: 148.7740368284103
# 0.4: 128.41293958030982
# 0.5: 112.38959614630065
# 0.6: 99.84492750001857


# EVALUATE

gamma = [0.1, 0.2, 0.8, 1.3, 1.5, 3, 5]

for g in gamma:

    image_path = f"./results/rice_gamma_{g}.png"

    image = load_image(image_path)

    plot_histogram(image, path=f"./hist/rice_gamma_{g}.png")


gamma = [0.1, 0.2, 0.3, 0.4, 0.5, 0.8]

for g in gamma:

    image_path = f"./results/plants_gamma_{g}.png"

    image = load_image(image_path)

    plot_histogram(image, path=f"./hist/plants_gamma_{g}.png")

```