Assignment 1

Ratchanon Panmas 6434480323

1. **Display the given images using 8, 64, 128, and 256 gray levels  
   Source Code**

import cv2

import numpy as np

import matplotlib.pyplot as plt

def display\_image(image, gray\_level):

    global image\_name

    gray\_image = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)

     # make it scaled to gray\_level

    gray\_image = np.floor(gray\_image / (256 // gray\_level)) \* (256 // gray\_level)

    cv2.imwrite(f"./images/quantized\_images/{image\_name}\_gray\_{gray\_level}.jpg", gray\_image)

    return gray\_image

images\_list = ["flower.jpg", "fractal.jpeg", "fruit.jpg"]

gray\_levels = [8, 16, 64, 128, 256]

image\_name = ""

# for image\_name in images\_list:

for image\_path in images\_list:

    image\_name = image\_path.split(".")[0]

    # plot configuration

    rows, columns = 1, len(gray\_levels) + 1

    fig = plt.figure(figsize=(30, 4))

    fig.suptitle(f"Image: {image\_path}", fontsize=15, y=1.05)

    # read image

    image = cv2.imread(f"images/{image\_path}")

    image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB) # convert to RGB

    # plot original image

    plt.subplot(rows, columns, 1)

    plt.imshow(image)

    plt.axis('off')

    plt.title(f"Original Image")

    for i in range(len(gray\_levels)):

        gray\_level = gray\_levels[i]

        grayscale\_img = display\_image(image, gray\_level)

        # plot quantized image

        plt.subplot(rows, columns, i + 2)

        plt.imshow(grayscale\_img, cmap='gray')

        plt.axis('off')

        plt.title(f"Gray Level: {gray\_level}")

    plt.savefig(f"images/quantized\_images/{image\_path}")

    plt.show()

A collage of different flowers

Description automatically generatedA screenshot of a computer generated image

Description automatically generatedA collage of different fruits

Description automatically generated

**ผลลัพธ์ที่ได้ คือ**

การแสดงภาพด้วยจำนวน Gray Level ที่แตกต่างกันนั้นให้คุณภาพของภาพที่แตกต่างกัน โดยที่จำนวน Gray Level 8 ระดับจะเห็นได้อย่างชัดเจนว่ารายละเอียดของภาพนั้นลดลง เมื่อเปรียบเทียบกับ Gray Level 64 ระดับ แต่หากเปรียบเทียบ Gray Level 64 128 และ 256 ระดับนั้นจะสังเกตถึงความแตกต่างของแต่ละระดับค่อนข้างยาก หากต้องการเปรียบเทียบควรดูรูป fruit.jpg จะเห็นได้ชัดว่าที่จำนวน Gray Level 8 level บริเวณกล้วยนั้นได้สูญเสียรายละเอียดไป จากการวิเคราะห์คิดว่าเนื่องจากบริเวณกล้วยมีความใกล้เคียงกันของสีค่อนข้างมากทำให้เมื่อลดจำนวน Gray Level ลงจึงเห็นความแตกต่างอย่างชัดเจนของการเปลี่ยนแปลงระดับค่าสีเทา

1. **Enhance the 8-bit gray scale images  
   Source Code**

import cv2

import matplotlib.pyplot as plt

def enhance\_img(image\_path):

image = cv2.imread(f"images/{image\_path}")

grayscale\_img = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

L = 256

for i in range(len(grayscale\_img)):

for j in range(len(grayscale\_img[0])):

if grayscale\_img[i][j] <= L / 3:

grayscale\_img[i][j] = 5 \* L / 6

elif grayscale\_img[i][j] <= 2 \* L / 3:

grayscale\_img[i][j] = (-2 \* grayscale\_img[i][j]) + 384

else:

grayscale\_img[i][j] = L / 6

cv2.imwrite(f"./images/enhanced\_images/{image\_path}", grayscale\_img)

return grayscale\_img

rows, columns = 1, 2

images\_list = ["flower.jpg", "traffic.jpg", "tram.jpg"]

for image\_path in images\_list:

fig = plt.figure(figsize=(30, 3))

enhance\_img(image\_path)

# plot original image

plt.subplot(rows, columns, 1)

image = cv2.imread(f"images/{image\_path}")

image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB) # convert to RGB

plt.imshow(image)

plt.axis('off')

plt.title(f"Original Image")

# plot enhanced image

plt.subplot(rows, columns, 2)

plt.imshow(cv2.imread(f"images/enhanced\_images/{image\_path}"), cmap='gray')

plt.axis('off')

plt.title(f"Enhanced Image")

plt.show()

**ผลลัพธ์ที่ได้คือ**

* ช่วง Gray Level 0 ถึง

ในช่วงนี้ผลลัพธ์ที่ได้คือ ในช่วงนี้จากบริเวณที่มีค่า Gray level จากมืดจะกลายเป็นสว่างที่ระดับ หากมี Gray level 256 ระดับจะมีค่า Gray level อยู่ที่ 213

* ช่วง Gray Level ถึง

ในช่วงนี้ผลลัพธ์ที่ได้คือ ในช่วงนี้จากบริเวณที่มีค่า Gray level จากมืดจะกลายเป็นสว่าง และบริเวณที่สว่างจะมืดลง โดยมีสมการเชิงเส้นการเปลี่ยนแปลงค่า Gray level เป็น

* ช่วง Gray Level ถึง

ในช่วงนี้ผลลัพธ์ที่ได้คือ ในช่วงนี้จากบริเวณที่มีค่า Gray level จากสว่างจะกลายเป็นมืดที่ระดับ หากมี Gray level 256 ระดับจะมีค่า Gray level อยู่ที่ 42.67 = 43

A close-up of a bunch of flowers

Description automatically generated A traffic jam in a city

Description automatically generated

A trolley on a street

Description automatically generated

1. **Enhance the 8-bit gray scale images using the power-law transformation**

**Source code**

import cv2

import numpy as np

import matplotlib.pyplot as plt

def power\_law\_transform(image\_path, c, gamma):

image = cv2.imread(image\_path)

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

image\_name = image\_path.split("/")[-1].split(".")[0]

gamma\_corrected\_img = np.array(255 \* (image / 255) \*\* gamma, dtype='uint8')

cv2.imwrite(f"./images/gamma\_corrected\_images/{image\_name}-{c}-{gamma}.jpg", gamma\_corrected\_img)

return gamma\_corrected\_img

images\_list = ["cartoon.jpg", "scenery1.jpg", "scenery2.jpg"]

for image\_path in images\_list:

path = f"images/{image\_path}"

fig = plt.figure(figsize=(7, 8))

row, col = 4, 2

# plot original image

plt.subplot(row, col, 1)

image = cv2.imread(path)

image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB) # convert to RGB

plt.imshow(image)

plt.axis('off')

plt.title(f"Original Image")

c\_list = [0.5, 1, 2]

gamma\_list = [0.4, 2.5]

for c in c\_list:

for gamma in gamma\_list:

plt.subplot(row, col, c\_list.index(c) \* 2 + gamma\_list.index(gamma) + 3)

plt.imshow(power\_law\_transform(path, c, gamma), cmap='gray')

plt.axis('off')

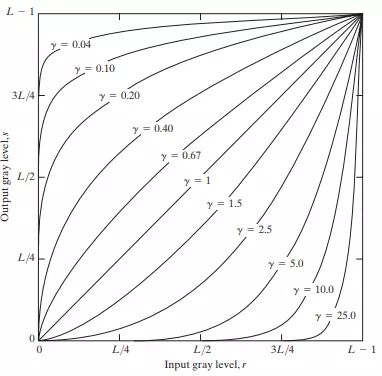
plt.title(f"c={c}, gamma={gamma}")

plt.savefig(f"images/gamma\_corrected\_images/{image\_path}")

plt.show()

**ผลลัพธ์ที่ได้คือ**

จากการทำ Power-law transformation จะสังเกตได้ว่าเมื่อค่า c เพิ่มขึ้นภาพที่ได้จะสว่างขึ้น ในส่วนของค่า gamma นั้นหากมีค่าน้อยกว่า 1 ภาพที่ได้จะสว่างขึ้น และเมื่อค่า gamma น้อยกว่า 1 จะทำให้ภาพมืดลงโดยสามารถอธิบายได้ด้วยกราฟ



A black and white image of a child's face

Description automatically generated A group of gray squares with text

Description automatically generated A group of images of trees

Description automatically generated