

```
import os
import cv2
import numpy as np

def map_to_gray_levels(image, levels):
    step = 256 // (levels)
    new_image = (image // step) * step
    return new_image.astype(np.uint8)

def main():
    asset_path = './asset'
    pictures = ["flower.jpg", "fractal.jpeg", "fruit.jpg"]

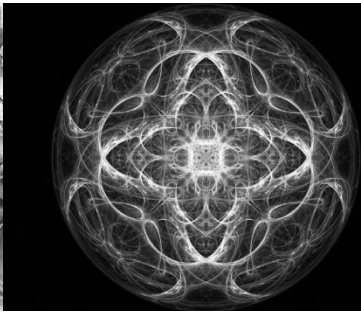
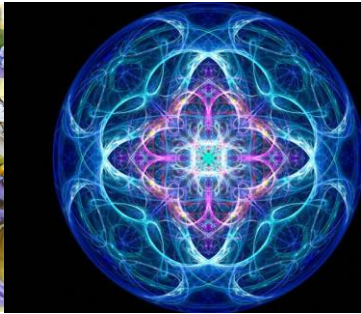
    for picture in pictures:
        image_path = os.path.join(asset_path, picture)
        image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
        if image is None:
            print("Could not open or find the image.")
            return

        gray8 = map_to_gray_levels(image, 8)
        gray64 = map_to_gray_levels(image, 64)
        gray128 = map_to_gray_levels(image, 128)
        gray256 = map_to_gray_levels(image, 256)
        pic_name = picture.split(".")[0]
        output_dir = f"./gray/{pic_name}"
        os.makedirs(output_dir, exist_ok=True)
        # Display images
        cv2.imshow(f'{pic_name}_original', image)
        cv2.imshow(f'{pic_name}_gray8', gray8)
        cv2.imshow(f'{pic_name}_gray64', gray64)
        cv2.imshow(f'{pic_name}_gray128', gray128)
        cv2.imshow(f'{pic_name}_gray256', gray256)
        # Save images
        cv2.imwrite(os.path.join(output_dir, f'{pic_name}_gray8.jpg'), gray8)
        cv2.imwrite(os.path.join(output_dir, f'{pic_name}_gray64.jpg'), gray64)
        cv2.imwrite(os.path.join(output_dir, f'{pic_name}_gray128.jpg'), gray128)
        cv2.imwrite(os.path.join(output_dir, f'{pic_name}_gray256.jpg'), gray256)

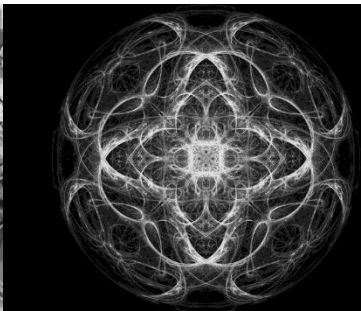
        cv2.waitKey(0)
        cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```

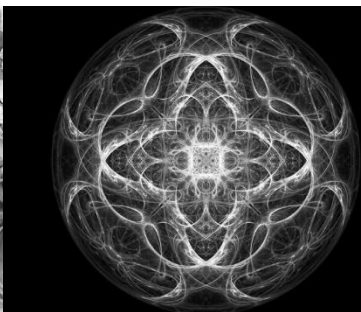
ภาพต้นฉบับ



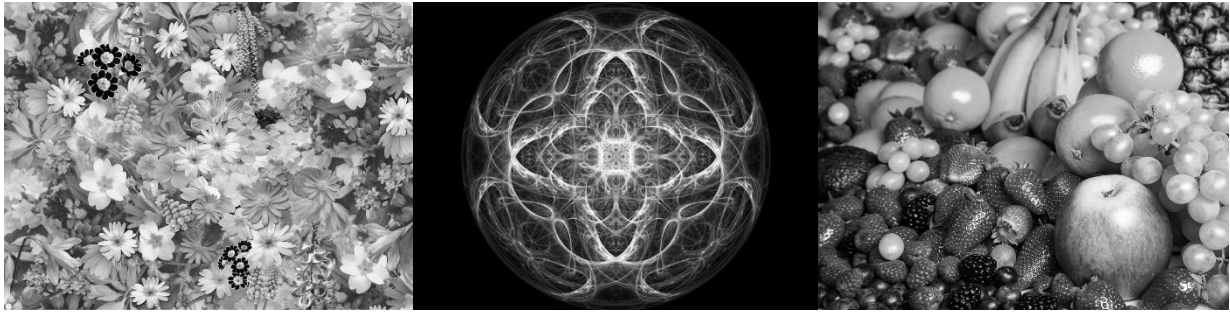
8 gray levels



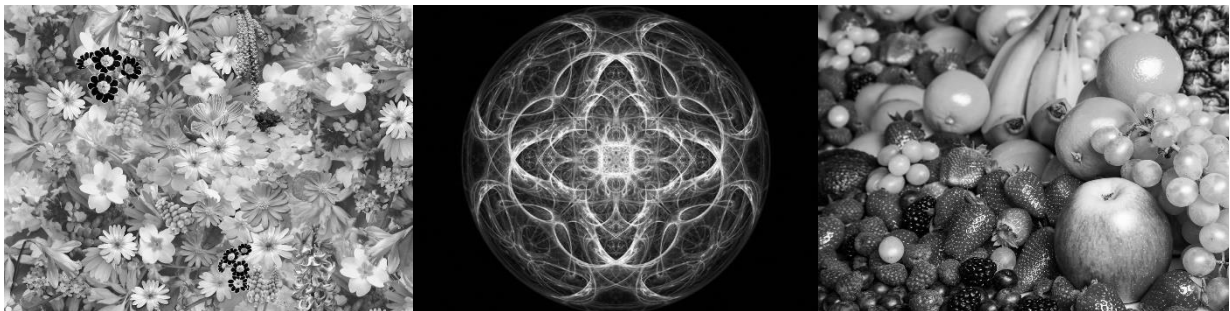
64 gray levels



128 gray levels



256 gray levels



อธิบายเพิ่มเติม : เมื่อปรับแต่งภาพแล้วโดยส่วนตัวจะเห็นความแตกต่างภาพ 8 gray levels โดยสามารถสังเกตได้คือบริเวณที่ใกล้เคียงกันจะมีสีโทนเดียวกันที่กระจุกอยู่ใกล้กัน เพราะบริเวณที่ใกล้กันจะอยู่ในช่วงของค่าใกล้เคียงกันจะทำให้ถูกปรับค่าและได้ค่าออกมาเป็นค่าเดียวกัน ส่วนภาพที่เหลือจะให้ความรู้สึกเหมือนกันภาพต้นฉบับ (ที่อ่านออกมาเป็น grayscale) เพราะถ้าดูจากค่าที่ทำการ print ออกมาแล้ว ค่าที่ได้ออกมาก็มีความใกล้เคียงกันจริงๆ เพราะ 8 gray levels มีช่วงของค่าที่มากที่สุดคือ 32 ที่จะถูกปรับไปเป็นค่าเดียวกัน ส่วน 64, 128, 256 gray levels มีช่วงที่จะปรับเพียง 4, 2, 1 ตามลำดับ และค่าที่ปรับออกมาก็จะมีค่าที่ไม่ต่างกันมาก ทำให้ผลที่ได้ไม่แตกต่างกันมาก

ข้อ 2

```
import cv2
import numpy as np
import os

def enhance_image(image):
    L = 256
    enhanced_image = np.copy(image)

    h, w = image.shape
```



```

    for y in range(h):
        for x in range(w):
            if 0 <= enhanced_image[y, x] < L/3:
                enhanced_image[y, x] = int(5 * L / 6)
            elif L/3 <= enhanced_image[y, x] < 2 * L/3:
                slope = ((L/6) - (5 * L/6)) / ((2 * L/3) - (L/3))
                enhanced_image[y, x] = int((5 * L/6) + (slope *
(enhanced_image[y, x] - (L/3))))
            else:
                enhanced_image[y, x] = int(L / 6)

    return enhanced_image

def main():
    asset_path = './asset'
    pictures = ["flower.jpg", "traffic.jpg", "tram.jpg"]
    for picture in pictures:
        image_path = os.path.join(asset_path, picture)
        image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
        if image is None:
            print("Could not open or find the image.")
            return

        enhanced_image = enhance_image(image)

        cv2.imshow("enhanced_image", enhanced_image)

        pic_name = picture.split(".")[0]
        output_dir = f"./enhanced/{pic_name}"
        os.makedirs(output_dir, exist_ok=True)
        cv2.imwrite(os.path.join(output_dir, f'{pic_name}_enhanced.jpg'),
enhanced_image)

        cv2.waitKey(0)
        cv2.destroyAllWindows()

if __name__ == '__main__':
    main()

```

ภาพต้นฉบับ



ภาพ enhance



อธิบายเพิ่มเติม : ภาพที่ได้จะมีลักษณะคล้ายกับ **negative** เพราะช่วงตั้งแต่ $L/3$ ถึง $2L/3$ มีลักษณะของ **slope** ที่ติดลบ คล้ายกับ **negative** เพียงแค่ค่าของ **slope** มีค่ามากกว่าทำให้การปรับเฉดมีการปรับที่ไวกว่า และส่วนที่แตกต่างอีกเล็กน้อยคือส่วนที่มีค่าน้อยกว่า $L/3$ และมากกว่า $2L/3$ ซึ่งจะแปลงค่าจากโทนดำเป็นโทนขาวและโทนขาวเป็นโทนดำตามลำดับ แต่จะแตกต่างกับส่วนที่เป็น **slope** ของฟังก์ชันคือค่าที่ได้จะเป็นค่าที่คงที่เลย

```

import cv2
import os
import numpy as np

def enhance_image(image, c, gamma):
    enhanced_image = np.power(image / 255.0, gamma)
    enhanced_image = np.uint8(255 * c * enhanced_image)
    return enhanced_image

def main():
    asset_path = './asset'
    pictures = ["cartoon.jpg", "scenery1.jpg", "scenery2.jpg"]
    for picture in pictures:
        image_path = os.path.join(asset_path, picture)

        image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
        if image is None:
            print("Could not open or find the image.")
            return

        c_values = [0.5, 1, 2]
        gamma_values = [0.4, 2.5]

        for c in c_values:
            for gamma in gamma_values:
                enhanced_image = enhance_image(image, c, gamma)
                cv2.imshow(f"enhanced_image c : {c} gamma : {gamma}",
enhanced_image)

                pic_name = picture.split(".")[0]
                output_dir = f"./enhanced_powerlaw/{pic_name}"
                os.makedirs(output_dir, exist_ok=True)
                cv2.imwrite(os.path.join(output_dir,
f'{pic_name}_c{c}_gamma{gamma}.jpg'), enhanced_image)

                cv2.waitKey(0)
                cv2.destroyAllWindows()

if __name__ == '__main__':
    main()

```

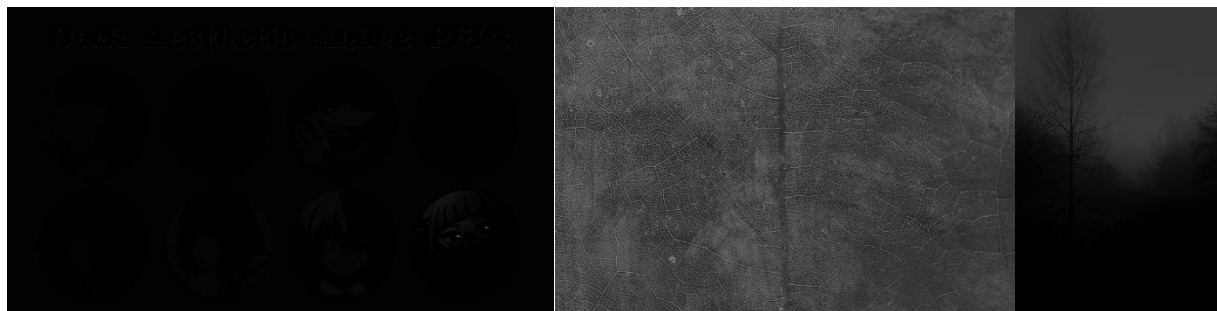
ภาพต้นฉบับ



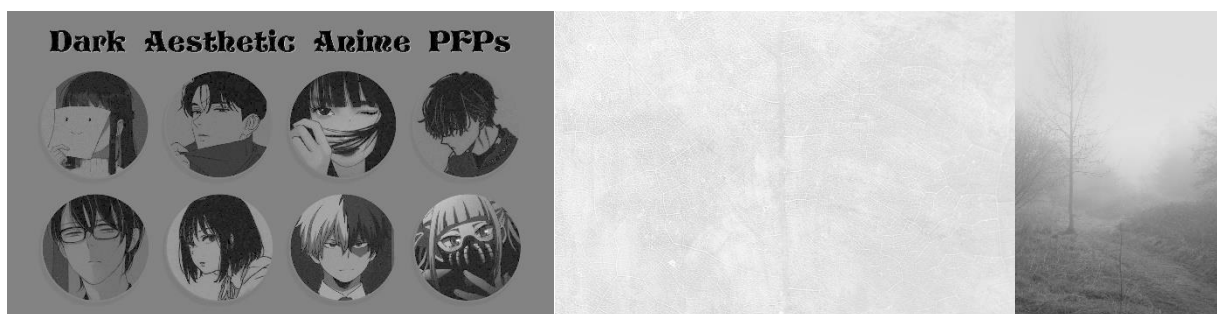
ภาพ enhance power-law $c = 0.5$ $\gamma = 0.4$



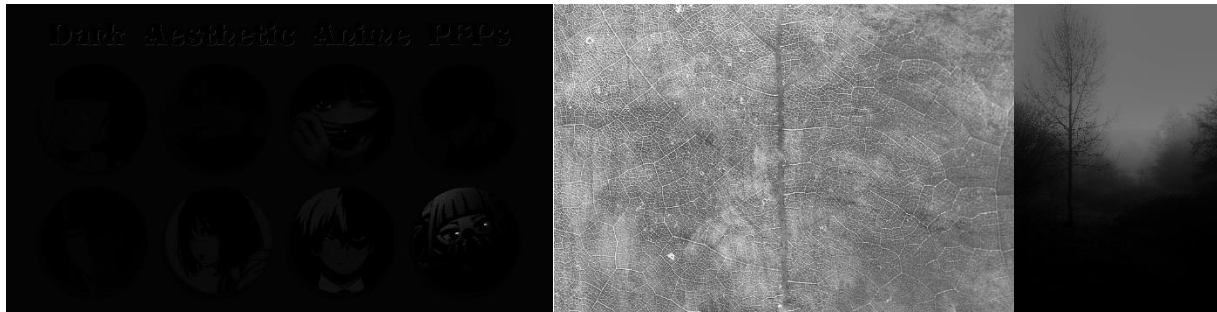
ภาพ enhance power-law $c = 0.5$ $\gamma = 2.5$



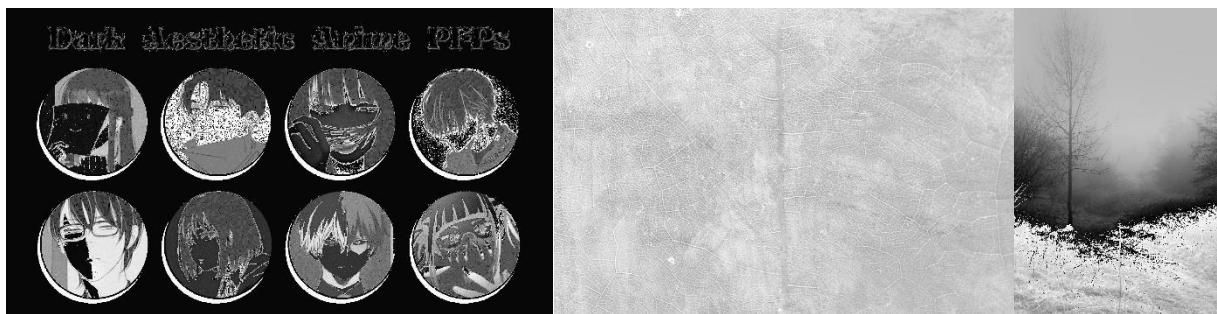
ภาพ enhance power-law $c = 1$ $\gamma = 0.4$



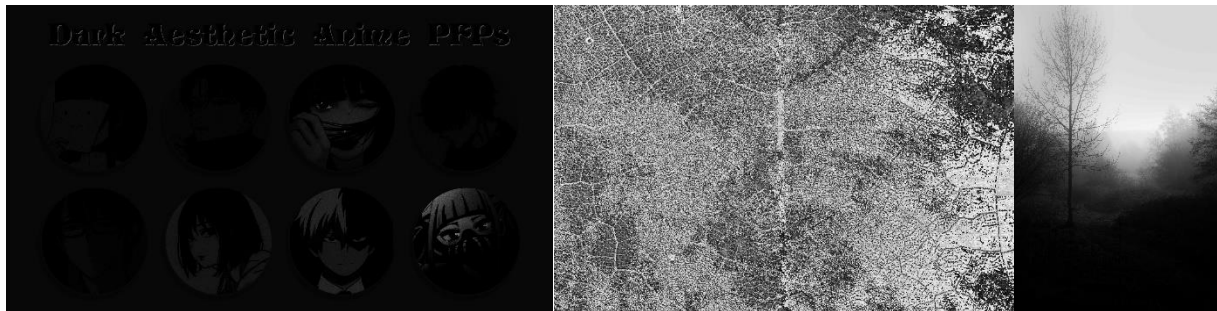
ภาพ enhance power-law $c = 1$ $\gamma = 2.5$



ภาพ enhance power-law $c = 2$ $\gamma = 0.4$



ภาพ enhance power-law $c = 2$ $\gamma = 2.5$



อธิบายเพิ่มเติม : ภาพที่ได้เมื่อ

$c = 0.5$ $\gamma = 0.4$ | ภาพที่ได้จะอยู่ในโทนดำกว่าภาพต้นฉบับ

$c = 0.5$ $\gamma = 2.5$ | ภาพที่ได้จะอยู่ในโทนดำกว่าภาพต้นฉบับ และก็โทนดำกว่า $c = 0.5$ $\gamma = 0.4$

$c = 1$ $\gamma = 0.4$ | ภาพที่ได้จะอยู่ในโทนขาวกว่าภาพต้นฉบับ

$c = 1$ $\gamma = 2.5$ | ภาพที่ได้จะอยู่ในโทนดำกว่าทั้งภาพต้นฉบับ และ $c = 1$ $\gamma = 0.4$

$c = 2$ $\gamma = 0.4$ | จะมีการแยกสีที่ชัดเจนกว่าภาพต้นฉบับส่วนที่ดำจะดำกว่า ส่วนที่ขาวก็จะขาวกว่า

$c = 2$ $\gamma = 2.5$ | ภาพที่ได้จะอยู่ในโทนต่ำกว่าภาพต้นฉบับ

โดยถ้าจะอธิบายภาพที่ได้จากสมการคือ ค่า c จะทำการปรับค่าของภาพให้เพิ่มลดได้โดย c เป็นค่าคงที่ กล่าวคือถ้าเพิ่มก็จะเพิ่มทั้งภาพถ้าลดก็จะลดทั้งภาพ ส่วนค่า γ จะทำให้ภาพสว่างขึ้นถ้าค่าน้อยกว่า 1 และจะทำให้ภาพดำนลงถ้าค่า γ มากกว่า 1