

Lab 5.1 กลุ่มงานบุนตึงตึง

1. display_resized_images Function:

```
def display_resized_images(image):
    # Define reduce factors and corresponding scale factors
    reduce_factors = [2, 5, 10]
    scale_factors = [1 / factor for factor in reduce_factors]

    # Define interpolation methods
    interpolation_methods = [cv2.INTER_NEAREST, cv2.INTER_LINEAR, cv2.INTER_CUBIC, cv2.INTER_AREA]
    str_interpolation_methods = ["INTER_NEAREST", "INTER_LINEAR", "INTER_CUBIC", "INTER_AREA"]

    plt.figure(figsize=(10, 5))

    for i, scale_factor in enumerate(scale_factors):
        for j, inter_method in enumerate(interpolation_methods):
            resized_image = cv2.resize(image, None, fx=scale_factor, fy=scale_factor, interpolation=inter_method)
            plt.subplot(len(scale_factors), len(interpolation_methods), i * len(interpolation_methods) + j + 1)
            plt.imshow(resized_image)
            plt.title(f"Scale Factor: {scale_factor}, Interpolation: {str_interpolation_methods[j]}", fontsize = 6)
            plt.xticks([]); plt.yticks([])

    plt.show()
```

ฟังก์ชันนี้ใช้สำหรับแสดงภาพที่ปรับขนาดและใช้วิธีการปรับขนาดต่าง ๆ บนกราฟ

```
# Define reduce factors and corresponding scale factors
reduce_factors = [2, 5, 10]
scale_factors = [1 / factor for factor in reduce_factors]
```

ตัวแปร reduce_factors คือระยะลดขนาดที่ต้องการ (2, 5, 10)

ตัวแปร scale_factors คือสัมประสิทธิ์การลดขนาดที่เป็นส่วนกลับของ reduce_factors

```
# Define interpolation methods
interpolation_methods = [cv2.INTER_NEAREST, cv2.INTER_LINEAR, cv2.INTER_CUBIC, cv2.INTER_AREA]
str_interpolation_methods = ["INTER_NEAREST", "INTER_LINEAR", "INTER_CUBIC", "INTER_AREA"]
```

ตัวแปร interpolation_methods เก็บวิธีการปรับขนาดที่มีให้ใน OpenCV

ตัวแปร str_interpolation_methods เก็บชื่อของวิธีการปรับขนาดเพื่อนำมาแสดงในกราฟ

2. อ่านภาพจากไฟล์:

```
# Read image file
image = cv2.imread("grid.jpg")
display_resized_images(image)
```

image จะเก็บภาพที่อ่านจากไฟล์ "grid.jpg" โดยใช้ cv2.imread() ซึ่งอ่านภาพจากไฟล์และเก็บเป็น numpy array ของค่าสี

3. เรียกใช้ display_resized_images:

```
display_resized_images(image)
```

โดยส่งภาพ image เข้าไปในฟังก์ชัน display_resized_images เพื่อแสดงผลลัพธ์

4. ในฟังก์ชัน display_resized_images:

```
plt.figure(figsize=(10, 5))

for i, scale_factor in enumerate(scale_factors):
    for j, inter_method in enumerate(interpolation_methods):
        resized_image = cv2.resize(image, None, fx=scale_factor, fy=scale_factor, interpolation=inter_method)
        plt.subplot(len(scale_factors), len(interpolation_methods), i * len(interpolation_methods) + j + 1)
        plt.imshow(resized_image)
        plt.title(f"Scale Factor: {scale_factor}, Interpolation: {str(interpolation_methods[j])}", fontsize = 6)
        plt.xticks([]); plt.yticks([])

plt.show()
```

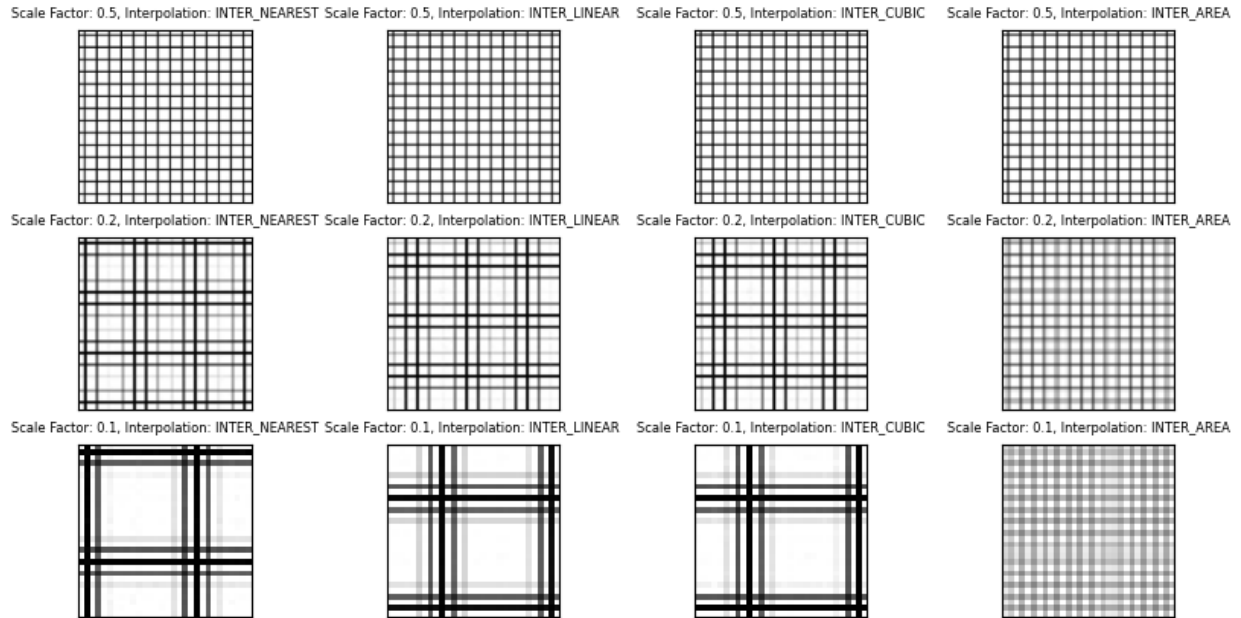
มีการวนลูปเพื่อสร้างกราฟที่มีแถวและคอลัมน์ตามจำนวนของ scale_factors และ interpolation_methods ด้วย matplotlib

สำหรับแต่ละครั้งที่วนลูปในกราฟ:

จะทำการปรับขนาดภาพต้นฉบับด้วย cv2.resize() โดยใช้ scale_factor และ inter_method ที่กำหนด

จากนั้นจะแสดงภาพที่ปรับขนาดแล้วด้วย matplotlib พร้อมกับคำอธิบายของขนาดและวิธีการปรับขนาด

ผลลัพธ์ที่ได้



ผลลัพธ์ที่ได้ประกอบด้วยรูปภาพที่แตกต่างกันที่มีการปรับขนาดและใช้วิธีการแปลงภาพต่าง ๆ ตามการกำหนดในโค้ด โดยแต่ละภาพจะมีคำบรรยายที่ระบุขนาดและวิธีการแปลงภาพที่ใช้ในการปรับขนาด.