

ข้อที่ 1 Smoothing filters to remove noise from the given images

- a.) Averaging filter
- b.) Median filter

Code

```
def apply_filter(image, filter_size, filter_function):  
    # Define the output image  
    output_image = np.copy(image)  
  
    # Calculate padding  
    padding = filter_size // 2  
  
    # Iterate over each pixel (excluding border pixels)  
    for i in range(padding, image.shape[0] - padding):  
        for j in range(padding, image.shape[1] - padding):  
            # Extract the neighborhood of the current pixel  
            window = image[i - padding:i + padding + 1, j - padding:j +  
padding + 1]  
            # Apply the filter function to the window and replace the current  
            # pixel value  
            output_image[i, j] = filter_function(window)  
  
    return output_image  
  
def apply_averaging_filter(image, filter_size=3):  
    return apply_filter(image, filter_size, np.mean)  
  
def apply_median_filter(image, filter_size=3):  
    return apply_filter(image, filter_size, np.median)
```

โดยที่เราจะใช้ filter ขนาด 3x3, 5x5, 7x7 และ 9x9 ได้ output ของภาพแรก noisy_img1.jpg ดังนี้

Original Image



Averaging Filter 3x3



Median Filter 3x3



Averaging Filter 5x5



Median Filter 5x5



Averaging Filter 7x7



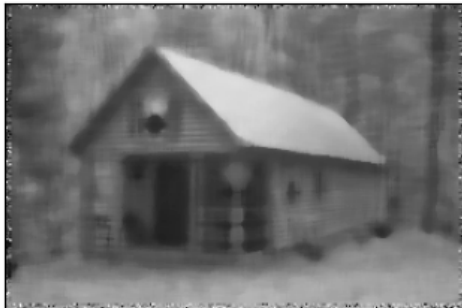
Median Filter 7x7



Averaging Filter 9x9



Median Filter 9x9



และภาพที่สอง noisy_img2.jpg

Original Image



Averaging Filter 3x3



Median Filter 3x3



Averaging Filter 5x5



Median Filter 5x5



Averaging Filter 7x7



Median Filter 7x7



Averaging Filter 9x9



Median Filter 9x9



เริ่มจากภาพแรก



จะสังเกตว่ามีsalt and pepper noise ซึ่งใน lecture เราบอกว่าการใช้ median filter จะได้ผลดีกว่ากับ noiseแบบนี้ ซึ่งจากผลลัพธ์พบว่าการใช้ median filter ได้ผลที่ดีกว่าการใช้ averaging filter จริงๆ โดยภาพที่ดีที่สุด คือการใช้ filter ขนาด3x3 ซึ่งให้ภาพที่ไม่เบลอจนเกินไป (ยิ่งขนาดfilterใหญ่ขึ้น ภาพที่ได้ยิ่งเบลอมากขึ้น) ดังภาพนี้



แต่ถ้าเราใช้ averaging filter ขนาดเท่ากัน 3x3 ภาพที่ได้จะออกมาเป็นแบบภาพข้างล่างนี้ ซึ่งจะเห็นว่ายังมองเห็นว่ามีnoiseอยู่



ในภาพที่สอง



มี salt and pepper noise เช่นเดียวกับภาพแรก แต่ดูเหมือนว่าในภาพนี้ noise จะมีความเยอะและละเอียดมากกว่า ภาพแรก ซึ่งการใช้ filter ขนาด 5×5 ดูจะดีที่สุดในการลบ noise ออกไปแต่ก็แน่นอนว่าภาพที่ได้จะค่อนข้างเบลอ ขอแสดงการใช้ filter ขนาด 5×5 ในการ enhancement

Averaging filter



Median filter



จะเห็นว่า Median filter ให้ผลลัพธ์ออกมาดีกว่าการใช้ filter แบบ averaging เพราะว่าภาพที่ได้จาก median filter มีความคมชัดมากกว่า ไม่เบลอมากเหมือน averaging filter สังเกตได้จากบริเวณขอบหลังคาบ้านของ median filter ที่มีความคมชัดมากกว่า

ทำให้เราอาจจะสรุปได้ว่าในภาพที่มี salt and pepper noise การใช้ Median filter ก็เป็นหนึ่งในทางเลือกที่ใช้ในการลบ noise ออกไปได้ดีกว่าการใช้ Averaging filter

ข้อที่ 2 Sharpening filters

a.) Laplacian filter

b.) Gradient filter

เนื่องจากโจทย์ไม่ได้บอกว่าใช้แบบไหน จึงจะทำทั้งหมด 4 แบบคือ Laplacian, Laplacian extension, Sobel Gradient และ Robert Gradient filter

Code

```
def apply_laplacian_filter(image):
    laplacian_kernel = np.array([
        [0, -1, 0],
        [-1, 5, -1],
        [0, -1, 0]
    ])
    sharpened = cv2.filter2D(image, -1, laplacian_kernel)
    return sharpened

def apply_laplacian_extension_filter(image):
    laplacian_kernel = np.array([
        [-1, -1, -1],
        [-1, 9, -1],
        [-1, -1, -1]
    ])
    sharpened = cv2.filter2D(image, -1, laplacian_kernel)
    return sharpened

def apply_sobel_gradient_filter(image):
    sobel_x = np.array([
        [-1, 0, 1],
        [-2, 0, 2],
        [-1, 0, 1]
    ])
    sobel_y = np.array([
        [-1, -2, -1],
        [0, 0, 0],
        [1, 2, 1]
    ])
    gradient_x = cv2.filter2D(image, -1, sobel_x)
    gradient_y = cv2.filter2D(image, -1, sobel_y)
    gradient_magnitude = np.sqrt(gradient_x**2 +
gradient_y**2).astype(np.uint8)
    sharpened = cv2.subtract(image, gradient_magnitude)
    return sharpened
```

```

def apply_robert_gradient_filter_sharpened(image):
    roberts_x = np.array([
        [0, 1],
        [-1, 0]
    ])
    roberts_y = np.array([
        [1, 0],
        [0, -1]
    ])
    gradient_x = cv2.filter2D(image, -1, roberts_x)
    gradient_y = cv2.filter2D(image, -1, roberts_y)
    gradient_magnitude = np.sqrt(gradient_x**2 +
gradient_y**2).astype(np.uint8)
    sharpened = cv2.subtract(image, gradient_magnitude)
    return sharpened

def apply_filter(image, filter_function):
    # Split the image into its RGB channels
    r, g, b = cv2.split(image)
    # Apply the filter function to each channel
    r_filtered = filter_function(r)
    g_filtered = filter_function(g)
    b_filtered = filter_function(b)
    # Merge the filtered channels
    filtered_image = cv2.merge([r_filtered, g_filtered, b_filtered])
    return filtered_image

# Load image
blurred_image = cv2.imread("../original_images/blurred_image.jpg")
# Apply the sharpening filters to the color image
laplacian = apply_filter(blurred_image, apply_laplacian_filter)
laplacian_extension = apply_filter(blurred_image,
apply_laplacian_extension_filter)
gradient_sobel = apply_filter(blurred_image,
apply_sobel_gradient_filter)
gradient_robert = apply_filter(blurred_image,
apply_robert_gradient_filter_sharpened)
# Save the output images
cv2.imwrite(f"./OUTPUT/laplacian.png", laplacian)
cv2.imwrite(f"./OUTPUT/laplacianExtension.png", laplacian_extension)
cv2.imwrite(f"./OUTPUT/gradientSobel.png", gradient_sobel)
cv2.imwrite(f"./OUTPUT/gradientRobert.png", gradient_robert)

```


ผลลัพธ์ที่ได้คือ

Original Image



Laplacian Filter



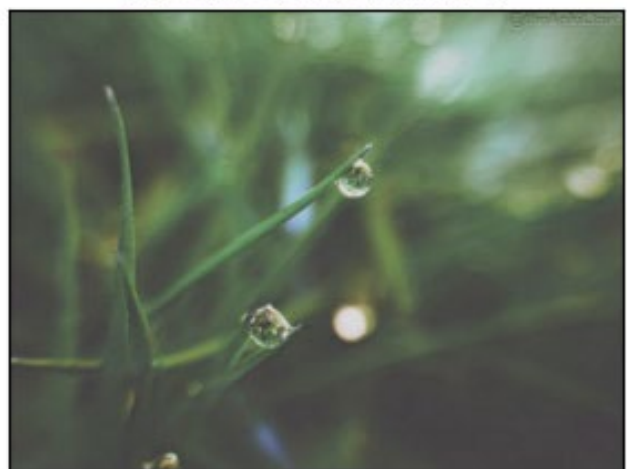
Laplacian Extension Filter



Sobel Gradient Filter

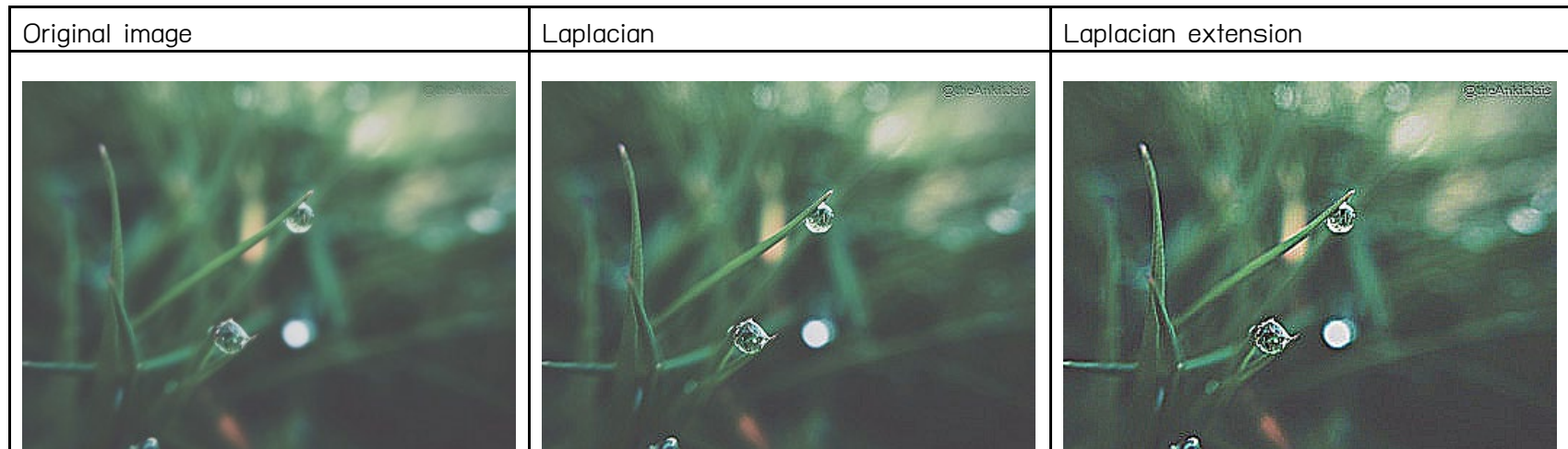


Robert Gradient Filter



จะเปรียบเทียบภาพที่ได้จากการทำ sharpening แบบต่างๆกับภาพต้นฉบับเป็นดังนี้

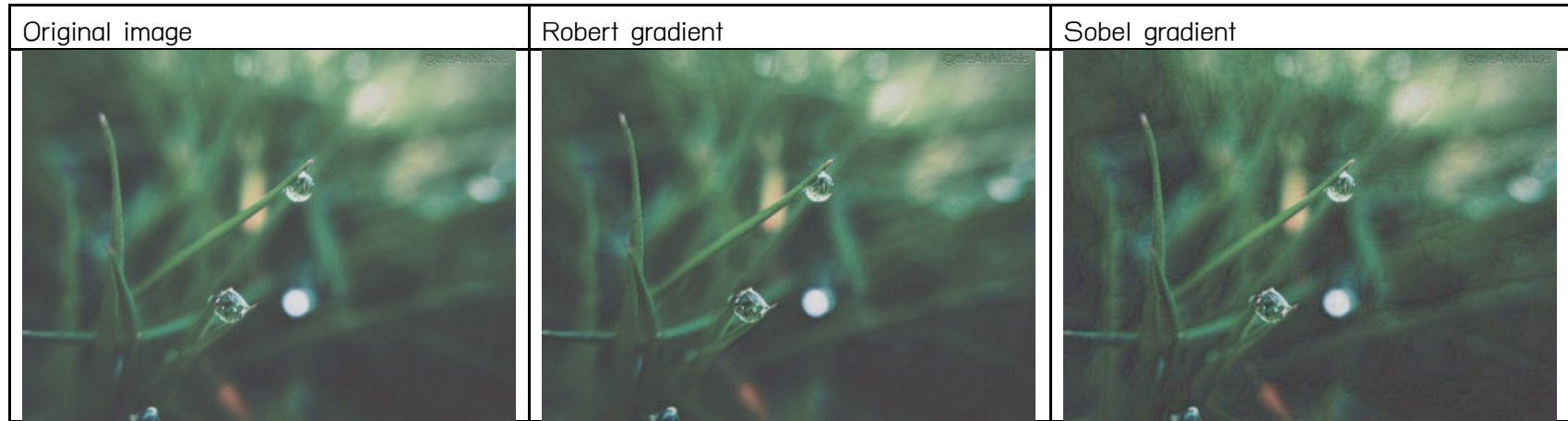
1. Laplacian



ในภาพที่ filter โดย Laplacian จะเห็นว่าความคมชัดบริเวณขอบและรายละเอียดในภาพดูมีความชัดเจนขึ้น ส่วนใน Laplacian extension ก็ยังมีความคมชัดในบริเวณขอบมากขึ้นไปอีก โดยถ้าเราสังเกตบริเวณหยดน้ำ เราจะสังเกตเห็นได้อย่างชัดเจนว่าภาพที่ใช้ Laplacian extension จะมีความคมชัดในบริเวณเส้นขอบมากที่สุด เนื่องจากการทำ Laplacian เป็นการทำให้เส้นขอบของภาพมีความเด่นชัดขึ้น



2. Gradient



จากภาพจะพบว่าภาพที่ได้จากการทำ gradient ในบริเวณพื้นหลังดูจะความคมชัดขึ้นเล็กน้อย ส่วนตัวใบไม้ และหยดน้ำก็มีความคมชัดเพิ่มขึ้นเล็กน้อยเท่านั้น แต่จะพบเห็นความแตกต่างในส่วนของสีมากกว่า โดยที่ gradient ทั้งแบบ Sobel และ Robert จะมีสีที่ดูเนียนและเข้มขึ้นกว่าภาพต้นฉบับ เมื่อเปรียบเทียบกันระหว่าง Sobel กับ Robert gradient จะพบว่าการใช้ Robert gradient ในภาพนี้ จะมีความคมชัดมากกว่าการใช้แบบ Sobel และสุดท้ายเมื่อเทียบ Gradient กับ Laplacian ภาพที่ได้มาจากการใช้ Laplacian ก็ดูจะมีความคมที่มากกว่าการใช้ Gradient