Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

Name	
Student ID	
Date	

Objectives

- Perform arithmetic operations (addition, subtraction, multiplication, and division) on images.
- Understand how these operations affect the intensity values of pixels.
- Explore how these operations are useful for basic image processing tasks.

Prerequisites

Make sure you have the following Python libraries installed:

- `opencv-python` (OpenCV)
- `matplotlib`

You can install them using following command in command prompt or bash:

pip install opencv-python, matplotlib

Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

Part 1: Arithmetic Operations on Images

Step 1: Subtracting an Intensity from an Image

Subtracting an intensity value from an image darkens it. You will subtract a constant value from all pixel intensities and observe the effect.

```
import cv2
image_color = cv2.imread('fruit basket.jpg')
image_sub = cv2.subtract(image_color, 100)
cv2.imshow('subtracted', image_sub)
cv2.waitKey(-1)
```

Or you may use the following source code to have a better comparison between the input and the output image

```
import cv2
import matplotlib.pyplot as plt

# Read a grayscale image
image = cv2.imread('fruit basket.jpg')
image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Subtract an intensity value from all pixels
subtracted_image = cv2.subtract(image_RGB, 100)

# Display original and subtracted images
plt.subplot(1, 2, 1)
plt.imshow(image_RGB)
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(subtracted_image)
plt.title('Subtracted Image (-100)')
```

Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

```
plt.axis('off')

plt.show()
```

Task 1:

1. Experiment with different intensity values (e.g., 30, 100) and observe how the image changes.
2. Explain why the image becomes darker as you subtract more.
3. The provided examples are for subtracting an intensity from all three RGB slices. Write a source code that reduce the intensities of the Red channel only.

Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

Step 2: Adding an Intensity to an Image

Adding an intensity value from an image brightens it. You will add a constant value to all pixel intensities and observe the effect.

```
import cv2
image_color = cv2.imread('fruit basket.jpg')
image_add = cv2.add(image_color, 100)
cv2.imshow('added', image_add)
cv2.waitKey(-1)
```

Or you may use the following source code to have a better comparison between the input and the output image

```
import cv2
import matplotlib.pyplot as plt

# Read a grayscale image
image = cv2.imread('fruit basket.jpg')
image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Add an intensity value to all pixels
added_image = cv2.add(image_RGB, 100)

# Display original and Added images
plt.subplot(1, 2, 1)
plt.imshow(image_RGB)
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(added_image)
plt.title('Added Image (100)')
plt.axis('off')

plt.show()
```

Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

Task 2:

4. Experiment with different intensity values (e.g., 30, 100) and observe how the image changes.
5. Explain why the image becomes brighter as you add more.
6. The provided examples are for adding an intensity from all three RGB slices. Write a source code that increases the intensities of the Red channel only.

Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

Step 3: Multiplying an Image by a Constant

Multiplying the pixel values by a constant can either increase or decrease the image contrast depending on the factor used.

```
import cv2
import matplotlib.pyplot as plt

# Read a grayscale image
image = cv2.imread('fruit basket.jpg')
image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Multiply an intensity value to all pixels
multiplied_image = cv2.multiply(image_RGB, 2)

# Display original and multiplied images
plt.subplot(1, 2, 1)
plt.imshow(image_RGB)
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(multiplied_image)
plt.title('Multiplied Image 2')
plt.axis('off')

plt.show()
```

Task 3:

1. Experiment with different multiplication factors (e.g., 0.5, 2.0) and observe how the image contrast changes.
2. Discuss the impact of multiplying the pixel values by a number greater than 1 and less than 1.

Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

Step 4: Dividing an Image by a Constant

Dividing the pixel values by a constant reduces the overall brightness of the image, similar to multiplying by a fraction.

```
import cv2
import matplotlib.pyplot as plt

# Read a grayscale image
image = cv2.imread('fruit basket.jpg')
image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Divide all pixels by an intensity value
multiplied_image = cv2.divide(image_RGB, 2)

# Display original and divided images
plt.subplot(1, 2, 1)
plt.imshow(image_RGB)
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(multiplied_image)
plt.title('Divided Image 2')
plt.axis('off')

plt.show()
```

Task 4:

1. Experiment with different division factors (e.g., 3.0, 0.5).
2. Compare the effects of dividing by a number greater than 1 versus less than 1.

Digital Image Processing
Spring 2026
Lab:2 Basic arithmetic operations on images

Submission

Submit the following:

1. Screenshots of the color, grayscale, and binary images you generated.
2. The code used for each step.
3. A brief report describing what you learned in this lab.

Conclusion

In this lab, you learned how to:

- Apply basic arithmetic operations (addition, subtraction, multiplication, and division) to images.
- Observe the effects of these operations on pixel intensities and overall image brightness and contrast.