**Image Arithmetics**

You can do some meaningful arithmetics on images to get various results. For example you can add images, subtract them, or even multiply them.

Now, we are going to test some of these mathematical operations.

In [ ]:

**from** skimage.io **import** imread

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

In [ ]:

live **=** imread('/content/0b37761a-de32-47ee-a3a4-e138b97ef542\_\_\_JR\_FrgE.S 2908.JPG')

mask **=** imread('/content/00e909aa-e3ae-4558-9961-336bb0f35db3\_\_\_JR\_FrgE.S 8593.JPG')

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

plt**.**subplot(121), plt**.**imshow(live, cmap**=**'gray')

plt**.**subplot(122), plt**.**imshow(mask, cmap**=**'gray')

plt**.**show()



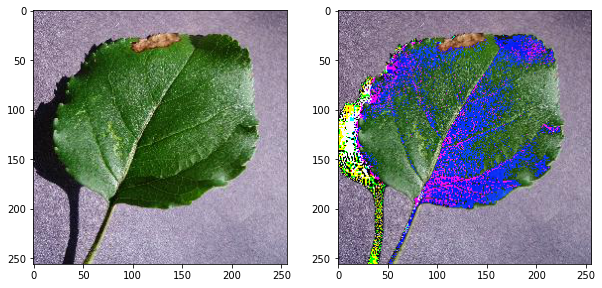
In [ ]:

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

plt**.**subplot(121), plt**.**imshow(live, cmap**=**'gray')

plt**.**subplot(122), plt**.**imshow(live **-** 20, cmap**=**'gray')

plt**.**show()



In [ ]:

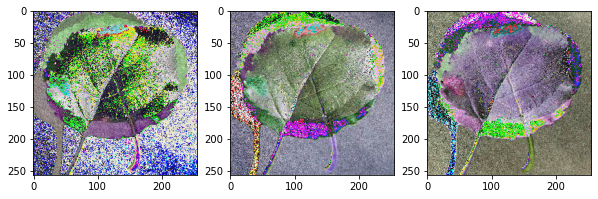
plt**.**figure(figsize**=**(10, 10))

plt**.**subplot(131), plt**.**imshow(mask **-** live, cmap**=**'gray')

plt**.**subplot(132), plt**.**imshow(**-**(mask **-** live **+** 128), cmap**=**'gray')

plt**.**subplot(133), plt**.**imshow(mask **-** live **+** 128, cmap**=**'gray')

plt**.**show()



In [ ]:

shaded **=** imread('/content/0b37761a-de32-47ee-a3a4-e138b97ef542\_\_\_JR\_FrgE.S 2908.JPG')

shading **=** imread('/content/00e909aa-e3ae-4558-9961-336bb0f35db3\_\_\_JR\_FrgE.S 8593.JPG')

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

plt**.**subplot(121), plt**.**imshow(shaded, cmap**=**'gray')

plt**.**subplot(122), plt**.**imshow(shading, cmap**=**'gray')

plt**.**show()



In [ ]:

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

plt**.**subplot(121), plt**.**imshow(np**.**multiply(shaded, 1**/**shading), cmap**=**'gray')

plt**.**subplot(122), plt**.**imshow(shaded, cmap**=**'gray')

plt**.**show()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:2: RuntimeWarning: divide by zero encountered in true\_divide

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:2: RuntimeWarning: invalid value encountered in multiply

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



In [ ]:

*# Test on the X-ray dental image*

xray **=** imread('/content/00e909aa-e3ae-4558-9961-336bb0f35db3\_\_\_JR\_FrgE.S 8593.JPG')

mask\_xray **=** imread('/content/00ddc106-692e-4c67-b2e8-569c924caf49\_\_\_Rutg.\_Bact.S 1228.JPG')

In [ ]:

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

plt**.**subplot(121), plt**.**imshow(xray, cmap**=**'gray')

plt**.**subplot(122), plt**.**imshow(mask\_xray, cmap**=**'gray')

plt**.**show()



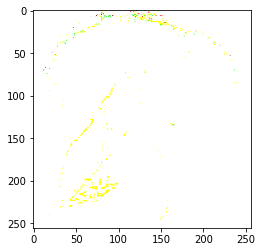
In [ ]:

plt**.**figure()

plt**.**imshow(np**.**multiply(xray, mask\_xray**/**255), cmap**=**'gray')

plt**.**show()

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



In [ ]:

*# Test on another image*

scan **=** imread('/content/0a62fe5a-22db-42e2-bca0-53a8dcfd8129\_\_\_RS\_NLB 0810.JPG')

print(scan**.**shape)

(256, 256, 3)

In [ ]:

*# Showing the body scan image*

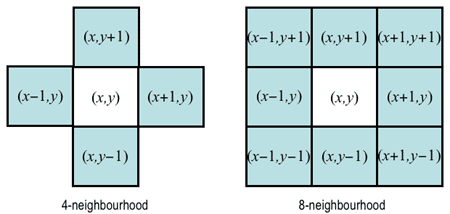
plt**.**figure(figsize**=**(7, 7))

plt**.**imshow(scan, cmap**=**'gray')

plt**.**show()



**Pixel relationships**



**Usual processes in DIP**

**Pixel (Point) processing**

Only individual pixels are entered into a process. The output is dependent on the single pixel values.

Some of this kind of processes are:

**Histogram Processing**

1. Contrast Enhancement
2. Histogram Equalization
3. Histogram Matching
4. Histogram Strtching

**Intensity Transformations**

1. Negative of an image
2. Log transformation
3. n-th power transformation
4. piecewise transformations

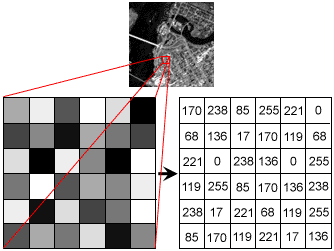
**Region (Neighborhood) processing**

A region (area) of pixels are entered into a process. The output is dependent on the values of entire region.

A common example of this kind of process includes **Spatial Filtering** and **Morphological Operators**, which are:

1. Average filtering
2. Median filtering
3. Sharpening filters
4. Edge detectors
5. Morphological Operations

**Histogram of an image**









In [ ]:

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

plt**.**subplot(211), plt**.**imshow(xray, cmap**=**'gray')

plt**.**subplot(212), plt**.**plot(np**.**histogram(xray, bins**=**256)[0])

plt**.**show()

