7.1 Contrast Enhancment

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

import cv2

import matplotlib.image as mpimg

from skimage import data

image = data.coffee()

image\_equalized = cv2.equalizeHist(image)

clahe = cv2.createCLAHE(clipLimit=2, tileGridSize=(8,8))

#Apply CLAHE to the original image

image\_clahe = clahe.apply(image)

image\_cs = np.zeros((image.shape[0],image.shape[1]),dtype = 'uint8')

# Apply Min-Max Contrasting

min = np.min(image)

max = np.max(image)

for i in range(image.shape[0]):

for j in range(image.shape[1]):

image\_cs[i,j] = 255\*(image[i,j]-min)/(max-min)

copyCamera = image.copy().astype(float)

m1,n1 = copyCamera.shape

output1 = np.empty([m1, n1])

for baris in range(0, m1-1):

for kolom in range(0, n1-1):

a1 = baris

b1 = kolom

output1[a1, b1] = copyCamera[baris, kolom] \* 1.9

fig, axes = plt.subplots(5, 2, figsize=(20, 20))

ax = axes.ravel()

ax[0].imshow(image, cmap=plt.cm.gray)

ax[0].set\_title("Citra Input")

ax[1].hist(image.ravel(), bins=256)

ax[1].set\_title('Histogram Input')

ax[2].imshow(image\_equalized, cmap=plt.cm.gray)

ax[2].set\_title("Citra Output HE")

ax[3].hist(image\_equalized.ravel(), bins=256)

ax[3].set\_title('Histogram Output HE Method')

ax[4].imshow(image\_cs, cmap=plt.cm.gray)

ax[4].set\_title("Citra Output CS")

ax[5].hist(image\_cs.ravel(), bins=256)

ax[5].set\_title('Histogram Output CS Method')

ax[6].imshow(image\_clahe, cmap=plt.cm.gray)

ax[6].set\_title("Citra Grayscale CLAHE")

ax[7].hist(image\_clahe.ravel(), bins=256)

ax[7].set\_title('Histogram Output CLAHE Method')

ax[8].imshow(output1, cmap=plt.cm.gray)

ax[8].set\_title("Citra Grayscale Perkalian Konstanta")

ax[9].hist(output1.ravel(), bins=256)

ax[9].set\_title('Histogram Output Perkalian Konstanta Method')

fig.tight\_layout()

7.2 Gray Level Slicing

import cv2

import numpy as np

from skimage import data

import matplotlib.pyplot as plt

%matplotlib inline

img = data.cat()

row, column = img.shape

img1 = np.zeros((row,column),dtype = 'uint8')

min\_range = 10

max\_range = 60

for i in range(row):

    for j in range(column):

        if img[i,j]>min\_range and img[i,j]<max\_range:

            img1[i,j] = 255

        else:

            img1[i,j] = 0

fig, axes = plt.subplots(2, 2, figsize=(12, 12))

ax = axes.ravel()

ax[0].imshow(img, cmap=plt.cm.gray)

ax[0].set\_title("Citra Input")

ax[1].hist(img.ravel(), bins=256)

ax[1].set\_title('Histogram Input')

ax[2].imshow(img1, cmap=plt.cm.gray)

ax[2].set\_title("Citra Output")

ax[3].hist(img1.ravel(), bins=256)

ax[3].set\_title('Histogram Output')

6.1 Filter Rerata

import matplotlib.pyplot as plt

%matplotlib inline

from skimage import data

from skimage.io import imread

from skimage.color import rgb2gray

import numpy as np

citra1 = imread(fname="mobil.tif")

citra2 = imread(fname="boneka2.tif")

print('Shape citra 1 : ', citra1.shape)

print('Shape citra 2 : ', citra2.shape)

fig, axes = plt.subplots(1, 2, figsize=(10, 10))

ax = axes.ravel()

ax[0].imshow(citra1, cmap = 'gray')

ax[0].set\_title("Citra 1")

ax[1].imshow(citra2, cmap = 'gray')

ax[1].set\_title("Citra 2")

copyCitra1 = citra1.copy().astype(float)

copyCitra2 = citra2.copy().astype(float)

m1,n1 = copyCitra1.shape

output1 = np.empty([m1, n1])

m2,n2 = copyCitra2.shape

output2 = np.empty([m2, n2])

print('Shape copy citra 1 : ', copyCitra1.shape)

print('Shape output citra 1 : ', output1.shape)

print('m1 : ',m1)

print('n1 : ',n1)

print()

print('Shape copy citra 2 : ', copyCitra2.shape)

print('Shape output citra 3 : ', output2.shape)

print('m2 : ',m2)

print('n2 : ',n2)

print()

for baris in range(0, m1-1):

for kolom in range(0, n1-1):

a1 = baris

b1 = kolom

jumlah = copyCitra1[a1-1, b1-1] + copyCitra1[a1-1, b1] + copyCitra1[a1-1, b1-1] + \

copyCitra1[a1, b1-1] + copyCitra1[a1, b1] + copyCitra1[a1, b1+1] + \

copyCitra1[a1+1, b1-1] + copyCitra1[a1+1, b1] + copyCitra1[a1+1, b1+1];

output1[a1, b1] = (1/9 \* jumlah)

fig, axes = plt.subplots(2, 2, figsize=(10, 10))

ax = axes.ravel()

ax[0].imshow(citra1, cmap = 'gray')

ax[0].set\_title("Input Citra 1")

ax[1].imshow(citra2, cmap = 'gray')

ax[1].set\_title("Input Citra 1")

ax[2].imshow(output1, cmap = 'gray')

ax[2].set\_title("Output Citra 1")

ax[3].imshow(output2, cmap = 'gray')

ax[3].set\_title("Output Citra 2")

6.2 Filter Median

import matplotlib.pyplot as plt

%matplotlib inline

from skimage import data

from skimage.io import imread

from skimage.color import rgb2gray

import numpy as np

citra1 = imread(fname="mobil.tif")

citra2 = imread(fname="boneka2.tif")

print('Shape citra 1 : ', citra1.shape)

print('Shape citra 1 : ', citra2.shape)

fig, axes = plt.subplots(1, 2, figsize=(10, 10))

ax = axes.ravel()

ax[0].imshow(citra1, cmap = 'gray')

ax[0].set\_title("Citra 1")

ax[1].imshow(citra2, cmap = 'gray')

ax[1].set\_title("Citra 2")

copyCitra1 = citra1.copy()

copyCitra2 = citra2.copy()

m1,n1 = copyCitra1.shape

output1 = np.empty([m1, n1])

m2,n2 = copyCitra2.shape

output2 = np.empty([m2, n2])

print('Shape copy citra 1 : ', copyCitra1.shape)

print('Shape output citra 1 : ', output1.shape)

print('m1 : ',m1)

print('n1 : ',n1)

print()

print('Shape copy citra 2 : ', copyCitra2.shape)

print('Shape output citra 3 : ', output2.shape)

print('m2 : ',m2)

print('n2 : ',n2)

print()

6.3 Filter Batas

citra1 = imread(fname="mobil.tif")

citra2 = imread(fname="boneka2.tif")

print('Shape citra 1 : ', citra1.shape)

print('Shape citra 1 : ', citra2.shape)

fig, axes = plt.subplots(1, 2, figsize=(10, 10))

ax = axes.ravel()

ax[0].imshow(citra1, cmap = 'gray')

ax[0].set\_title("Citra 1")

ax[1].imshow(citra2, cmap = 'gray')

ax[1].set\_title("Citra 2")

copyCitra1 = citra1.copy()

copyCitra2 = citra2.copy()

m1,n1 = copyCitra1.shape

output1 = np.empty([m1, n1])

m2,n2 = copyCitra2.shape

output2 = np.empty([m2, n2])

print('Shape copy citra 1 : ', copyCitra1.shape)

print('Shape output citra 1 : ', output1.shape)

print('m1 : ',m1)

print('n1 : ',n1)

print()

print('Shape copy citra 2 : ', copyCitra2.shape)

print('Shape output citra 3 : ', output2.shape)

print('m2 : ',m2)

print('n2 : ',n2)

print()

Konvolusi

import matplotlib.pyplot as plt

%matplotlib inline

from skimage import data

from skimage.io import imread

from skimage.color import rgb2gray

import numpy as np

import cv2

citra1 = imread(fname="gedung.tif")

print(citra1.shape)

plt.imshow(citra1, cmap='gray')

kernel = np.array([[-1, 0, -1],

[0, 4, 0],

[-1, 0, -1]])

citraOutput = cv2.filter2D(citra1, -1, kernel)

fig, axes = plt.subplots(1, 2, figsize=(12, 12))

ax = axes.ravel()

ax[0].imshow(citra1, cmap = 'gray')

ax[0].set\_title("Citra Input")

ax[1].imshow(citraOutput, cmap = 'gray')

ax[1].set\_title("Citra Output")