

-> ID: 20180261
-> ID : 20180252

```
# import libs #
import cv2
import numpy as np
from matplotlib import pyplot as plt

# read image #
img = cv2.imread("kernel.png", cv2.IMREAD_COLOR)

# converted it to 2D Grey image #
grey_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# write new image #
cv2.imwrite("filter-imgs/kernel-grey.png", grey_image)

# show plotting For The Grey Image #
plt.hist(grey_image)
plt.title("Grey Image Histogram")
plt.xlabel("greyscale value")
plt.ylabel("freq")
plt.figure()
plt.show()

# Display the new grey image #
cv2.imshow("Grey Image", grey_image)
cv2.waitKey(0)

# ----- Gaussian Noise ----- #

gauss = np.random.normal(0, .5, grey_image.size)
gauss = gauss.reshape(grey_image.shape[0],
```

```

grey_image.shape[1]).astype('uint8')

# Adding Gaussian Noise #
gaussian_img = cv2.add(grey_image, gauss)

# write new image #
cv2.imwrite("filter-imgs/kernel-grey-gaussian-noise.png", gaussian_img)

# Display the new Gaussian Noise Image #
cv2.imshow("Gaussian Noise Grey Image", gaussian_img)
cv2.waitKey(0)

# ----- Smoothing Filters ----- #

# ----- Average Filter ----- #

# read image #
average_image = cv2.imread("kernel.png", cv2.IMREAD_COLOR)
average_image = cv2.blur(average_image, (49, 49))

# write new image #
cv2.imwrite("filter-imgs/kernel-average-filter.png", average_image)

# Display the new Average Filter Image #
cv2.imshow("Average Filter Image", average_image)
cv2.waitKey(0)

# ----- Median Filter ----- #

# read image #
median_image = cv2.imread("kernel.png", cv2.IMREAD_COLOR)
# converted it to 2D Grey image #
grey_median_image = cv2.cvtColor(median_image, cv2.COLOR_BGR2GRAY)

median_image_filter = cv2.medianBlur(grey_median_image, 9)

# write new image #
cv2.imwrite("filter-imgs/kernel-median-filter.png", median_image_filter)

# Display the new Average Filter Image #
cv2.imshow("Median Filter Image", median_image_filter)
cv2.waitKey(0)

# ----- Sharpening Filters ----- #

# ----- Sobel Filter ----- #

# read image #
sobel_image = cv2.imread("kernel.png", cv2.IMREAD_COLOR)

# converted it to 2D Grey image #

```

```

grey_sobel_image = cv2.cvtColor(sobel_image, cv2.COLOR_BGR2GRAY)

sobelX = cv2.Sobel(grey_sobel_image, cv2.CV_64F, 1, 0, ksize=3)
sobelY = cv2.Sobel(grey_sobel_image, cv2.CV_64F, 0, 1, ksize=3)
sobel_image_filter = sobelX + sobelY

# Display Images #
cv2.imshow('Image sobel X filter', sobelX)
cv2.imshow('Image sobel y filter', sobelY)
cv2.imshow('Image sobel filter', sobel_image_filter)
cv2.waitKey(0)

# write new image #
cv2.imwrite("filter-imgs/kernel-sobel-x-filter.png", sobelX)
cv2.imwrite("filter-imgs/kernel-sobel-y-filter.jpg", sobelY)
cv2.imwrite("filter-imgs/kernel-sobel-filter.jpg", sobel_image_filter)
cv2.waitKey(0)

# ----- Prewitt Filter ----- #

# read image #
prewiit_image = cv2.imread("kernel.png", cv2.IMREAD_COLOR)

# converted it to 2D Grey image #
grey_prewiit_image = cv2.cvtColor(prewiit_image, cv2.COLOR_BGR2GRAY)

prewiit_x = np.array([[1,1,1],[0,0,0],[-1,-1,-1]])
prewiit_y = np.array([[-1,0,1],[-1,0,1],[-1,0,1]])

prewiit_image_filter = cv2.filter2D(grey_prewiit_image, -1, prewiit_x)
prewiit_image_filter = cv2.filter2D(grey_prewiit_image, -1, prewiit_y)

# Display Prewitt Image #
cv2.imshow('Image Prewitt filter', prewiit_image_filter)

# write new image #
cv2.imwrite("filter-imgs/kernel-prewiit-filter.jpg", prewiit_image_filter)

print("\033[95m"+"[*] The output Images from filters is in `filter-imgs` directory")

```