

# 1 Digital Image Filtering Using OpenCV

**Topic:** OpenCV is a very powerful tool for computer vision and image processing tasks. It provides various functions that allow us to perform easily useful tasks such as filtering an image.

**Exercise:** Read an image from disk and apply the following digital image processing filters:

- Mean filter of size:  $3 \times 3$ ,
- Gaussian mask,
- Vertical and horizontal edge detection filters,
- Laplacian filter,
- Laplacian of Gaussian filter.

## 2 Solution

```
import cv2
import numpy as np
from scipy import signal

# read image
image = cv2.imread("./image.jpg")
cv2.imshow("Original", image)
cv2.waitKey(0)

'''
negative
black: 0 (negative)-> 255
white: 255 (negative)-> 0
pixel (negative)-> 255 - value(pixel)
'''
image_negative = 255 * np.ones(image.shape) - image
cv2.imshow("Negative", image_negative.astype(np.uint8))
cv2.waitKey(0)

# create a mean filter mask
mean = np.ones((3, 3))
mean = (1/np.sum(mean)) * mean
# filter the image with the mean mask
```

```

image_mean = cv2.filter2D(image, -1, mean).astype(np.uint8)
cv2.imshow("Mean", image_mean)
cv2.waitKey(0)

# create gaussian mask
gauss = np.array([[1, 2, 1], [2, 4, 2], [1, 2, 1]])
gauss = (1/np.sum(gauss)) * gauss
# filter the image with the gaussian mask
image_gauss = cv2.filter2D(image, -1, gauss).astype(np.uint8)

cv2.imshow("Image Gauss", image_gauss)
cv2.waitKey(0)

# create a mask for vertical edge detection
vertical = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]])
# create a mask for horizontal edge detection
horizontal = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]])

# detect vertical edges
image_ver = cv2.filter2D(image, -1, vertical).astype(np.uint8)
# detect horizontal edges
image_hor = cv2.filter2D(image, -1, horizontal).astype(np.uint8)
# create a Laplacian mask
laplacian = np.array([[0, -1, 0], [-1, 4, -1], [0, -1, 0]])
# create Laplacian of Gaussian mask
lapofgaus = signal.convolve2d(laplacian, gauss)
# apply Laplacian
image_lap = cv2.filter2D(image, -1, laplacian).astype(np.uint8)
# apply LoG
image_lapog = cv2.filter2D(image, -1, lapofgaus).astype(np.uint8)
# for visualization purposes
image_lapog = 2*image_lapog
cv2.imshow("horizontal", image_hor)
cv2.imshow("vertical", image_ver)
cv2.imshow("laplacian", image_lap)
cv2.imshow("lapog", image_lapog)
cv2.waitKey(0)
cv2.destroyAllWindows()

```