Name: Aakash Kale
Class: BE Comps D
Roll No: CIA707
Batch: D1

EXPERIMENT NO: 1

Aim: Implement Image Processing methods e.g. Point Operator, Linear filtering etc.

Theory:

1. Point Operators:

Point operators, also known as pixel-wise operations, are image processing techniques that modify the pixel values of an image without considering the surrounding pixels. These operations are applied independently to each pixel, making them simple but powerful for various tasks like contrast adjustment, brightness correction, and enhancing specific image features.

2. Linear Filtering:

Linear filtering involves applying a convolution operation with a filter/kernel to an image. These filters are matrices that define the weights used in the convolution, and they are designed to extract specific features, blur, sharpen, or enhance an image.

Code:
Linear Filtering:

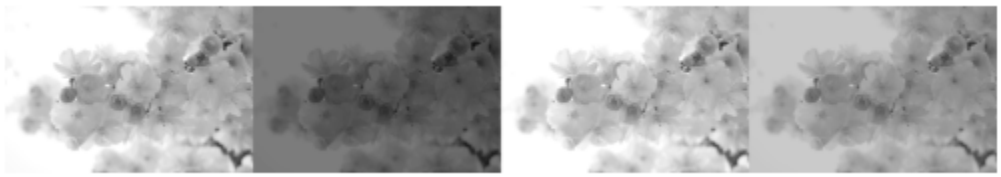
```
from google.colab.patches import cv2_imshow
import cv2 as cv
img = cv.imread("/content/pink-324175_1280.jpg")

cv2_imshow(img)
cv.waitKey(0)
import numpy as np
import matplotlib.pyplot as plt
import cv2
def point_operation(img, K, L):
  img = np.asarray(img, dtype=np.float)
  img = img*K + L
  # clip pixel values
  img[img > 255] = 255
```

```
  img[img < 0] = 0
  return np.asarray(img, dtype = np.int)
def main():
  # read an image
  img = cv2.imread('/content/pink-324175_1280.jpg')
  gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
  # k = 0.5, l = 0
  out1 = point_operation(gray, 0.5, 0)
  # k = 1., l = 10
  out2 = point_operation(gray, 1., 10)
  # k = 0.8, l = 15
  out3 = point_operation(gray, 0.7, 25)
  res = np.hstack([gray,out1, out2, out3])
  plt.imshow(res, cmap='gray')
  plt.axis('off')
  plt.show()
if __name__ == '__main__':
  main()
```

Input:

Output:



Box Filtering:

```
def plot_cv_img(input_image, output_image):
  fig, ax = plt.subplots(nrows=1, ncols=2)
  ax[0].imshow(cv2.cvtColor(input_image, cv2.COLOR_BGR2RGB))
  ax[0].set_title('Input Image')
  ax[0].axis('off')
  ax[1].imshow(cv2.cvtColor(output_image, cv2.COLOR_BGR2RGB))
  ax[1].set_title('Box Filter (50,50)')
  ax[1].axis('off')
  plt.show()
def main():
  # read an image
  img = cv2.imread('/content/pink-324175_1280.jpg')
  # To try different kernel, change size here.
  kernel_size = (50,50)
  # opencv has implementation for kernel based box blurring
  blur = cv2.blur(img,kernel_size)
  # Do plot
  plot_cv_img(img, blur)
if __name__ == '__main__':
  main()
```

Input and Output: