

# Assignment 6

## Image Filtering with Custom `filter2D`

**Md.Al-Amin Babu**  
ID: **2110676134**

### 1 Kernels

The following  $3 \times 3$  kernels were used:

#### Average

$$K_{\text{avg}} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

#### Sobel

$$K_{\text{Sx}} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_{\text{Sy}} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

#### Prewitt

$$K_{\text{Px}} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_{\text{Py}} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

#### Scharr

$$K_{\text{Schx}} = \begin{bmatrix} -1 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix}, \quad K_{\text{Schy}} = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix}$$

#### Laplacian

$$K_{\text{Lap}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

#### Custom Kernels

$$K_1 = \begin{bmatrix} -1 & 2 & 0 \\ 5 & -4 & 2 \\ 0 & 1 & -2 \end{bmatrix}, \quad K_2 = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 0 & 1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$K_3 = \begin{bmatrix} 1 & -2 & 0 \\ 0 & -1 & 1 \\ 2 & 0 & 2 \end{bmatrix}, \quad K_4 = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 3 \\ -2 & -1 & -2 \end{bmatrix}$$

## 2 How it Works

Each output pixel is computed by sliding a kernel over the image and taking the sum of elementwise products between the kernel and the corresponding image region:

$$O(i, j) = \sum_{u=0}^{k_h-1} \sum_{v=0}^{k_w-1} I(i+u, j+v) K(u, v).$$

Two modes are used:

- **same**: output size equals input (zero padding).
- **valid**: only fully-supported regions (smaller output).

## 3 Code Repository

The complete implementation is available at:  
<https://github.com/Al-Amin134>

## 4 Results

Below are the two terminal runs and their produced images. All image files must be placed *in the same folder* as this .tex file: image1.png, output.1.png, image2.png, output.2.png.

```
alamin@alaminbabu210:~/1.PART_IV/DIP/DIP_basic(lec 1)/online classes$ python filter2D.py
Enter the mode (must same or valid) : same
main Image height: 280 width: 260
laplace Image height: 280 width: 260
█
```

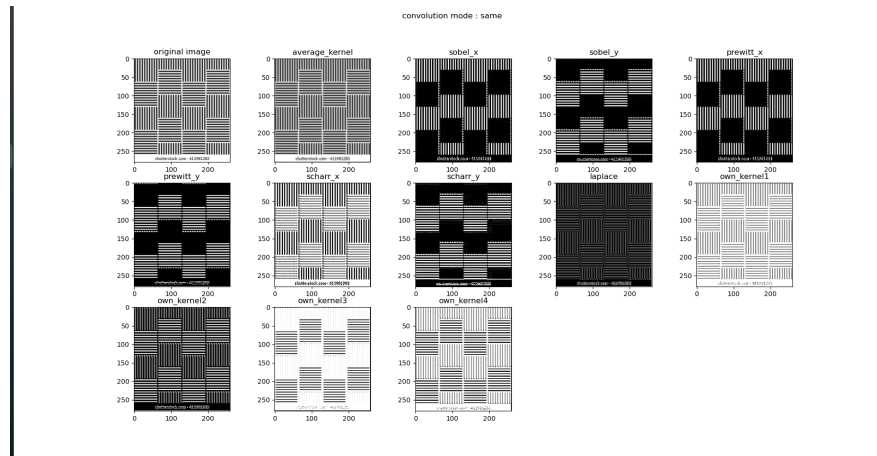


Figure 1: Convolution Mode : Same.

```

alamin@alaminbabu210:~/1.PART_IV/DIP/DIP_basic(lec 1)/online classes$ python filter2D.py
Enter the mode (must same or valid) : same
main Image height: 280 width: 260
laplace Image height: 280 width: 260
alamin@alaminbabu210:~/1.PART_IV/DIP/DIP_basic(lec 1)/online classes$ python filter2D.py
Enter the mode (must same or valid) : valid
main Image height: 280 width: 260
laplace Image height: 278 width: 258

```

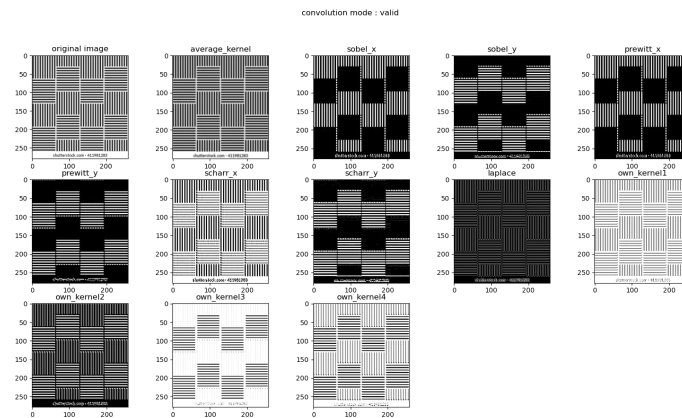


Figure 2: Convolution Mode : Valid.