

Machine Vision Lab 4 ¶

22BAI1037

Gorav Jhabakh

```
In [2]: import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [6]: img= cv2.imread("image.jpg")
img.shape
```

```
Out[6]: (3403, 5266, 3)
```

```
In [7]: plt.imshow(img)
plt.axis('off')
```

```
Out[7]: (np.float64(-0.5), np.float64(5265.5), np.float64(3402.5), np.float64(-0.5))
```



```
In [17]: gray_img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)  
plt.imshow(gray_img, cmap='gray')  
plt.axis('off')
```

```
Out[17]: (np.float64(-0.5), np.float64(5265.5), np.float64(3402.5), np.float64(-0.  
5))
```



```
In [18]: gray_img = cv2.resize(gray_img, (155,100))  
plt.imshow(gray_img, cmap='gray')  
plt.axis('off')
```

```
Out[18]: (np.float64(-0.5), np.float64(154.5), np.float64(99.5), np.float64(-0.5))
```



Gaussian Filter

```
In [33]: def gaussianFilter(img):
        kernel = np.array([[1, 2, 2, 1],
                           [2, 4, 4, 2],
                           [2, 4, 4, 2],
                           [1, 2, 2, 1]])/ 36.0

        rows,cols = img.shape
        padded = np.pad(img, pad_width=2, mode='constant', constant_values=0)
        output = np.zeros_like(img)

        for i in range(rows):
            for j in range(cols):
                box = padded[i:i+4, j:j+4]
                output[i,j] = np.sum(box*kernel)
        return output
```

```
In [34]: gaussian_img = gaussianFilter(gray_img)

plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.imshow(gray_img, cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(gaussian_img, cmap='gray')
plt.axis('off')
plt.title('Grayscale image with Gaussian filter')
plt.show()
```

Grayscale Image



Grayscale image with Gaussian filter



Min Filter

```
In [35]: def minFilter(img):

        padded = np.pad(img, pad_width=1, mode='constant', constant_values=0)
        output = np.zeros_like(img)
        rows,cols = img.shape

        for i in range(rows):
            for j in range(cols):
                box = padded[i:i+3, j:j+3]
                output[i,j] = np.min(box)
        return output
```

```
In [36]: min_img = minFilter(gray_img)

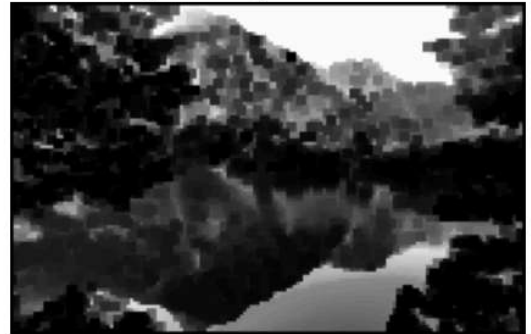
plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.imshow(gray_img, cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(min_img, cmap='gray')
plt.axis('off')
plt.title('Grayscale image with Min filter')
plt.show()
```

Grayscale Image



Grayscale image with Min filter



Max filter

```
In [37]: def maxFilter(img):

    padded = np.pad(img, pad_width=1, mode='constant', constant_values=0)
    output = np.zeros_like(img)
    rows,cols = img.shape

    for i in range(rows):
        for j in range(cols):
            box = padded[i:i+3, j:j+3]
            output[i,j] = np.max(box)
    return output
```

```
In [38]: max_img = maxFilter(gray_img)

plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.imshow(gray_img, cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(max_img, cmap='gray')
plt.axis('off')
plt.title('Grayscale image with Max filter')
plt.show()
```

Grayscale Image



Grayscale image with Max filter



```
In [39]: def medianFilter(img):

    padded = np.pad(img, pad_width=1, mode='constant', constant_values=0)
    output = np.zeros_like(img)
    rows,cols = img.shape

    for i in range(rows):
        for j in range(cols):
            box = padded[i:i+3, j:j+3]
            output[i,j] = np.median(box)
    return output
```



```
In [40]: median_img = medianFilter(gray_img)

plt.figure(figsize=(12,6))
plt.subplot(1,2,1)
plt.imshow(gray_img, cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(median_img, cmap='gray')
plt.axis('off')
plt.title('Grayscale image with Median filter')
plt.show()
```

Grayscale Image



Grayscale image with Median filter



Inference

- Gaussian Filter: The image is smoother and less noisy but the edges aren't sharp
- Min Filter: The image is darker and small bright spots are removed
- Max Filter: The image is brighter and small dark spots are removed
- Median Filter: The image is cleaner with very less noise but the edges are sharp