Department of Computer Science and Engineering

Submitted By:

Student Id:	C181208
Name:	Sameha Hasan
Section:	8AF
Course Code:	CSE-4875
Course Title:	Pattern Recognition and Image Processing sessional
Email:	samehasan25@gmail.com

Submitted To:

Mr. Mohammad Mahadi Hassan

Associate Professor,

Dept. of CSE , IIUC.

Lab 07:Smoothing filtering

- **Lowpass filtering**
- **Median Filtering**
- **Max filtering**
- **Min filtering**
- **Midpoint filtering**

jupyter C181208 Lab7 IP Last Checkpoint: 18 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

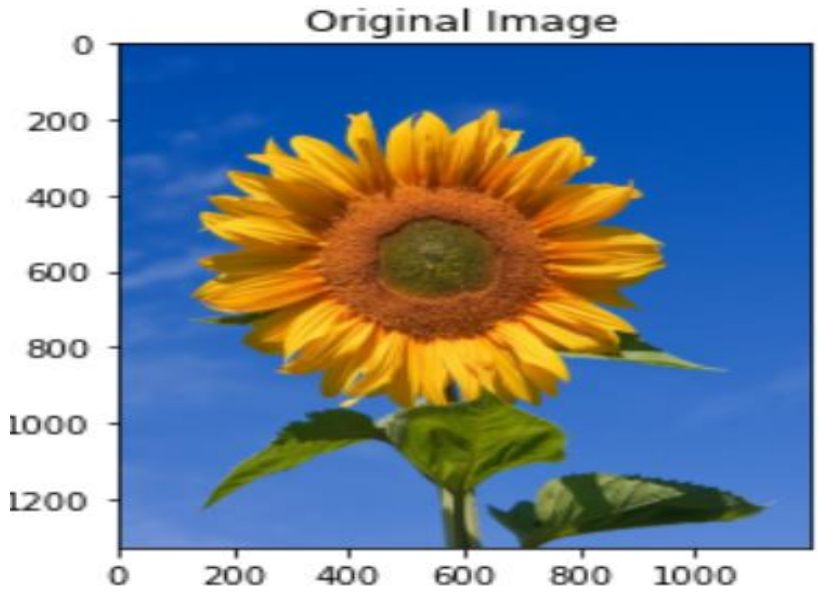
Importing necessary libraries and functions

In [1]:
```
from PIL import Image, ImageFilter
import cv2
import numpy as np
import matplotlib.pyplot as plt
from scipy import ndimage
%matplotlib inline
```

Reading necessary images

In [2]:
```
img = cv2.imread('Sunflower.jpg',1)
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
```

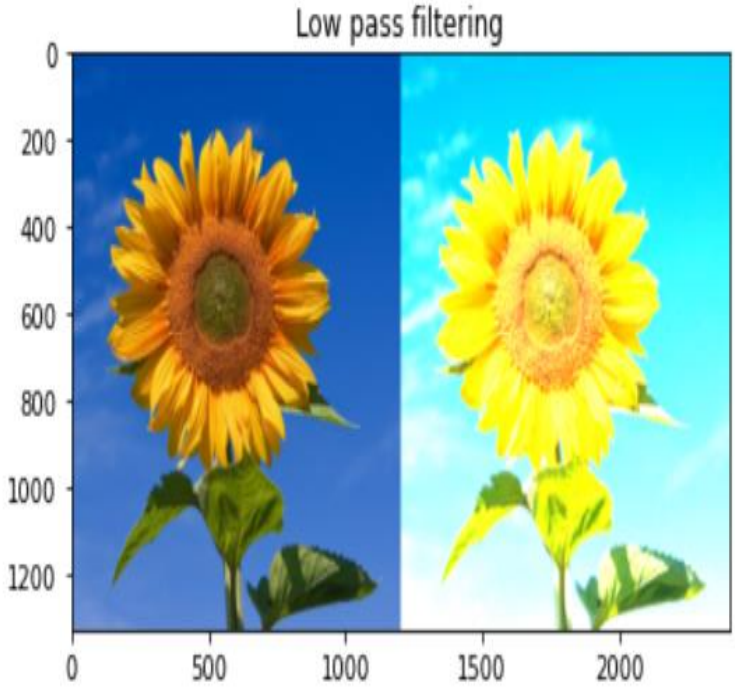
In [5]:
```
plt.imshow(img)
plt.title('Original Image')
img_data=np.array(img)
```



Task #1: Low Pass Filtering

In [6]:
```
kernel_low =np.array([[1/9,1/9,1/9,1/9,1/9],
[1/9,1/9,1/9,1/9,1/9],
[1/9,1/9,1/9,1/9,1/9],
[1/9,1/9,1/9,1/9,1/9],
[1/9,1/9,1/9,1/9,1/9]])
lowpass_5=cv2.filter2D(src=img,kernel=kernel_low,ddepth=-1) #used a mask size of 5x5
out_img=np.hstack((img_data,lowpass_5))
plt.imshow(out_img)
plt.title('Low pass filtering')
```

Out[6]:
```
Text(0.5, 1.0, 'Low pass filtering')
```

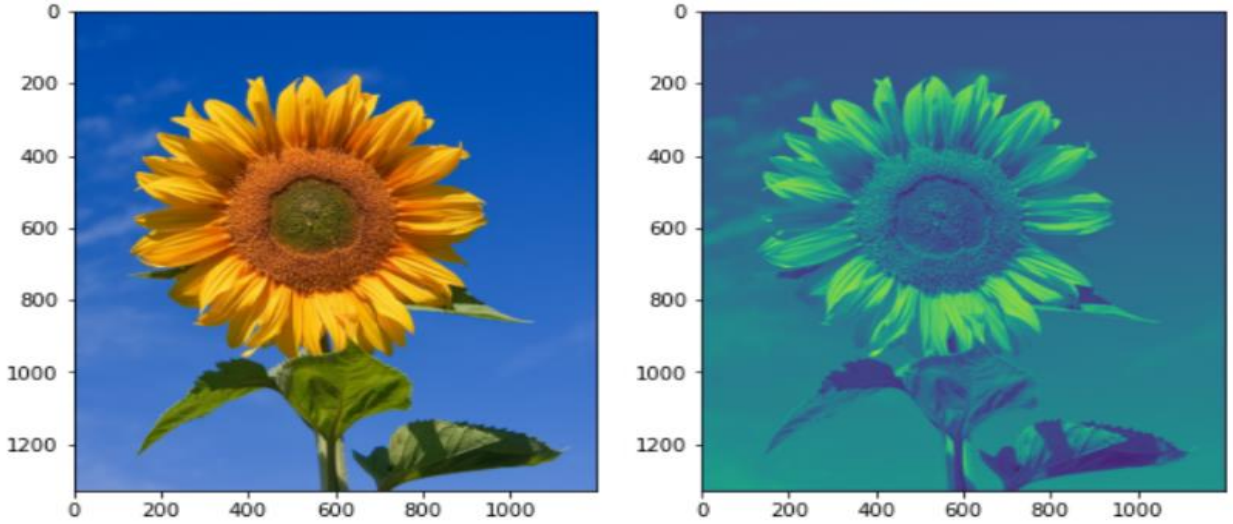


Task #2: Median Filtering

```
img_noisy1 = cv2.imread('Sunflower.jpg', 0)

# Obtain the number of rows and columns
# of the image
m, n = img_noisy1.shape

# Traverse the image. For every 3X3 area,
# find the median of the pixels and
# replace the center pixel by the median
img_new1 = np.zeros([m, n])

for i in range(1, m-1):
    for j in range(1, n-1):
        temp = [img_noisy1[i-1, j-1],
                img_noisy1[i-1, j],
                img_noisy1[i-1, j + 1],
                img_noisy1[i, j-1],
                img_noisy1[i, j],
                img_noisy1[i, j + 1],
                img_noisy1[i + 1, j-1],
                img_noisy1[i + 1, j],
                img_noisy1[i + 1, j + 1]]

        temp = sorted(temp)
        img_new1[i, j]= temp[4]
```

```
img_new1 = img_new1.astype(np.uint8)
cv2.imwrite('new_median_filtered.jpg', img_new1)
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.imshow(img)
plt.subplot(1,2,2)
plt.imshow(img)
plt.imshow(img_new1)
```

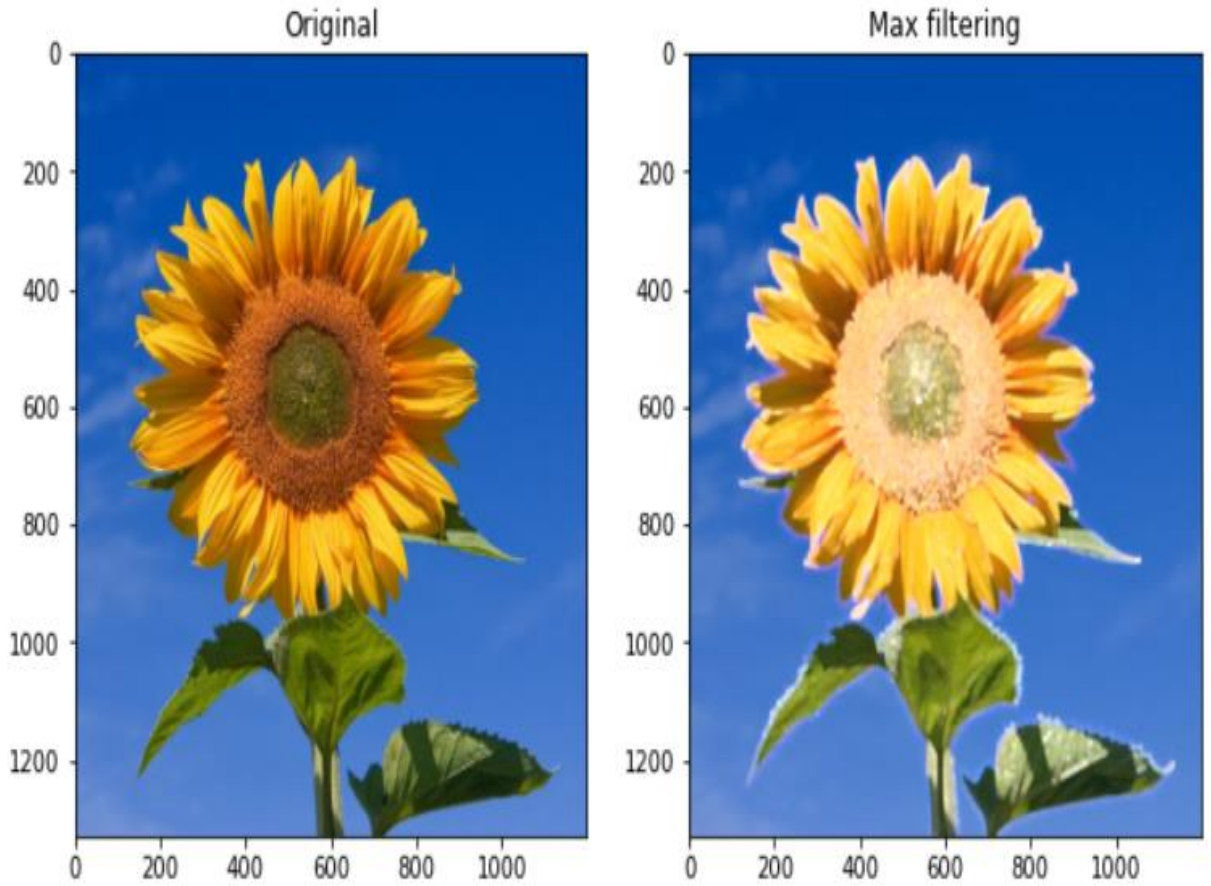
Out[7]: <matplotlib.image.AxesImage at 0x23d8966b940>



Task #3: Max Filtering

In [10]:
```
img3 = Image.open('Sunflower.jpg')
img4 = img3.filter(ImageFilter.MaxFilter(size=9)) #used a mask size of 9x9
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.imshow(img3)
plt.title('Original')
plt.subplot(1,2,2)
plt.imshow(img4)
plt.title('Max filtering')
```

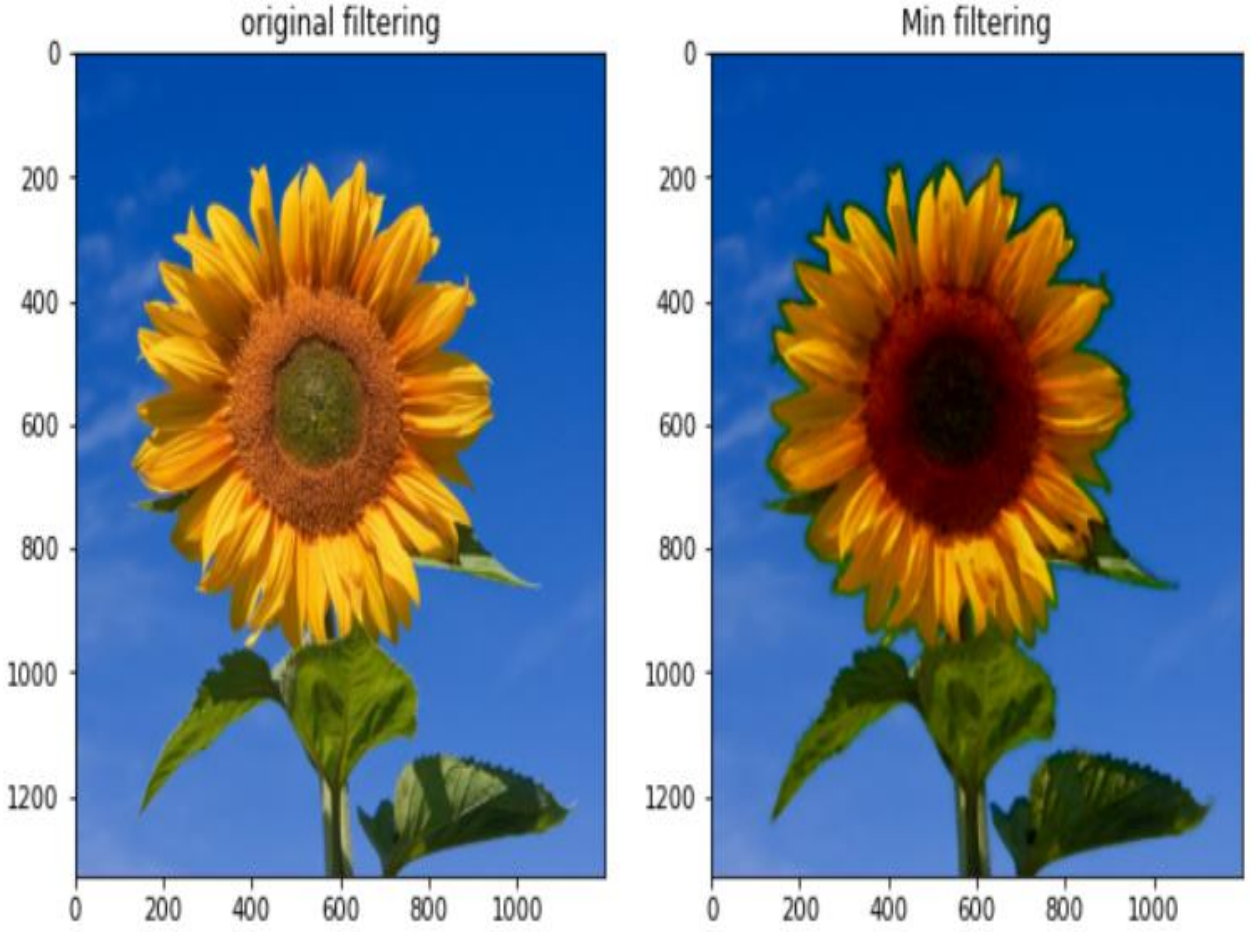
Out[10]: Text(0.5, 1.0, 'Max filtering')



Task #4: Min Filtering

In [14]:
```
img1 = Image.open('Sunflower.jpg')
img6 = img1.filter(ImageFilter.MinFilter(size=9))
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.imshow(img5)
plt.title('original filtering')
plt.subplot(1,2,2)
plt.imshow(img6)
plt.title('Min filtering')
```

Out[14]: Text(0.5, 1.0, 'Min filtering')



Task #5: Midpoint Filtering

In [15]:
```
img7 = Image.open('Sunflower.jpg')
img8 = img7.filter(ImageFilter.MinFilter(size=9))
img9 = img7.filter(ImageFilter.MaxFilter(size=9))
from numpy import asarray
img8=asarray(img8)
img9=asarray(img9)
mid_img=(img8+img9)*(1/2)
plt.imshow(np.uint8(mid_img))
plt.title('Midpoint filtering')
```

Out[15]:
```
Text(0.5, 1.0, 'Midpoint filtering')
```