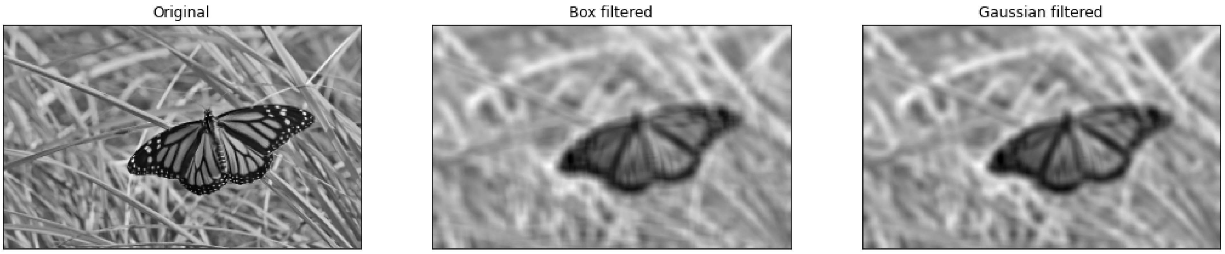
Pankajan T. 190428D

In [ ]:

```
%matplotlib inline
import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:

```
#Q1

img = cv.imread('butterfly.jpg', cv.IMREAD_REDUCED_GRAYSCALE_4)

# Box filter for averaging
box = 1./81.*np.ones((9,9))
img_box = cv.filter2D(img,-1,box)


# Box filter
gaus = cv.getGaussianKernel(9, 4)
img_gaus = cv.sepFilter2D(img, -1, gaus, gaus)


fig, axes  = plt.subplots(1,3, sharex='all', sharey='all', figsize=(18,18))



axes[0].imshow(img, cmap='gray')
axes[0].set_title('Original')
axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(img_box, cmap='gray')
axes[1].set_title('Box filtered')
axes[1].set_xticks([]), axes[1].set_yticks([])
axes[2].imshow(img_gaus, cmap='gray')
axes[2].set_title('Gaussian filtered')
axes[2].set_xticks([]), axes[1].set_yticks([])



plt.show()
```

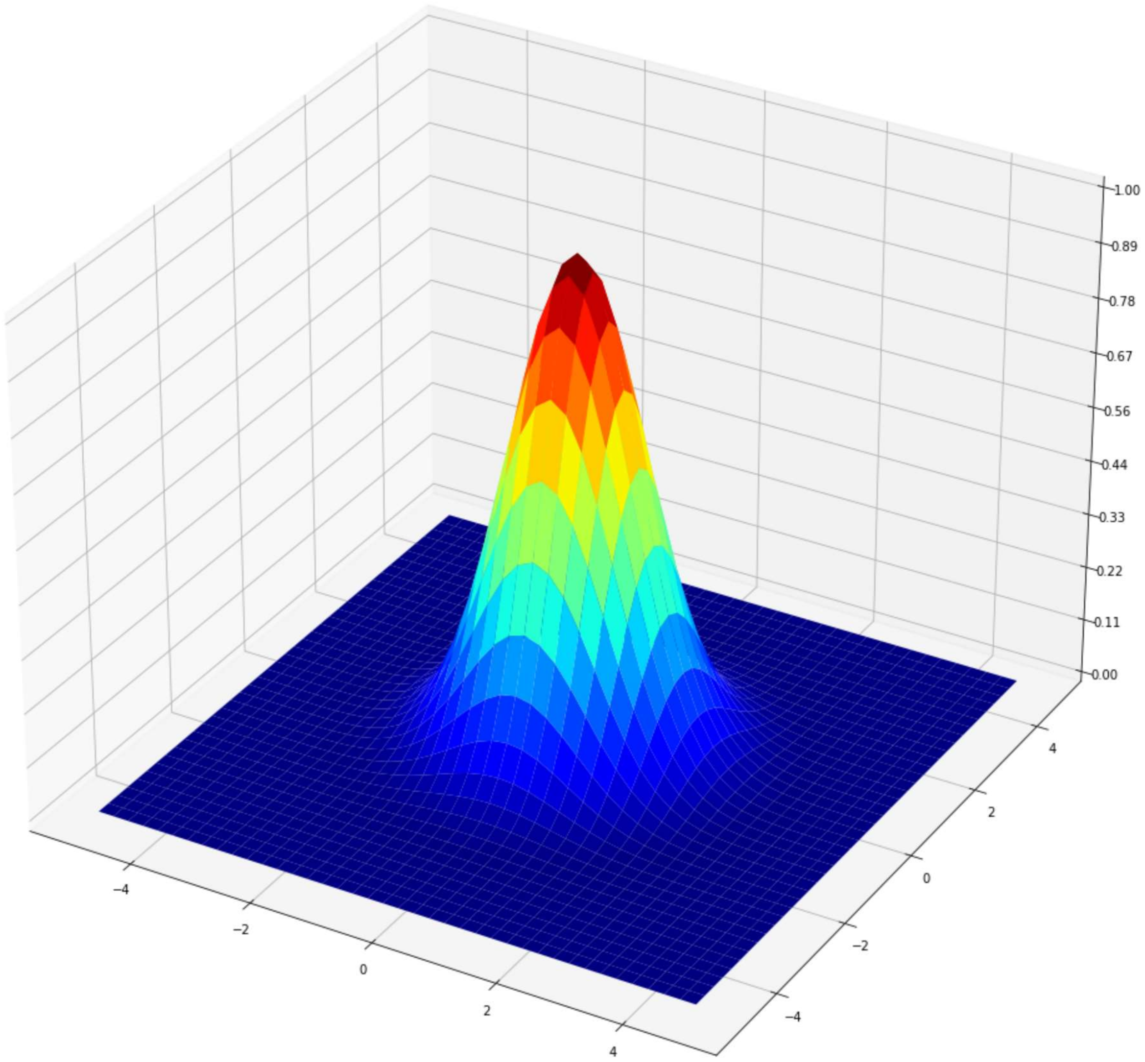


In [ ]:

```
#Q2



from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
import numpy as np
```

```
fig = plt.figure(figsize=(18,18))
ax = fig.add_subplot(111, projection='3d')
sigma = 1


X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
Z = np.exp(-(X**2 + Y**2)/(2*sigma**2))

# Plot the surface.
surf = ax.plot_surface(X, Y, Z, cmap=cm.jet, linewidth=0, antialiased=True)

# Customize the z axis.


#ax.set_zlim(-1.01, 1.01)
ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))
ax.set_aspect('equal', 'box')


#ax.view_init(90, 0)

cset = ax.contourf(X, Y, Z, zdir='z', offset=np.min(Z) -1.5, cmap=cm.jet)
ax.set_zlim(np.min(Z) - 2, np.max(Z))
# Add a color bar which maps values to colors.
#fig.colorbar(surf, shrink=0.5, aspect=5)
# Hide grid lines
# ax.grid(False)
plt.axis('off')

# Hide axes ticks
# ax.set_xticks([])
# ax.set_yticks([])
# ax.set_zticks([])

# plt.savefig('../../EN2550Lectures/en2550_lec03_spatial_filtering/figures/gaussian_2d

plt.show()
```

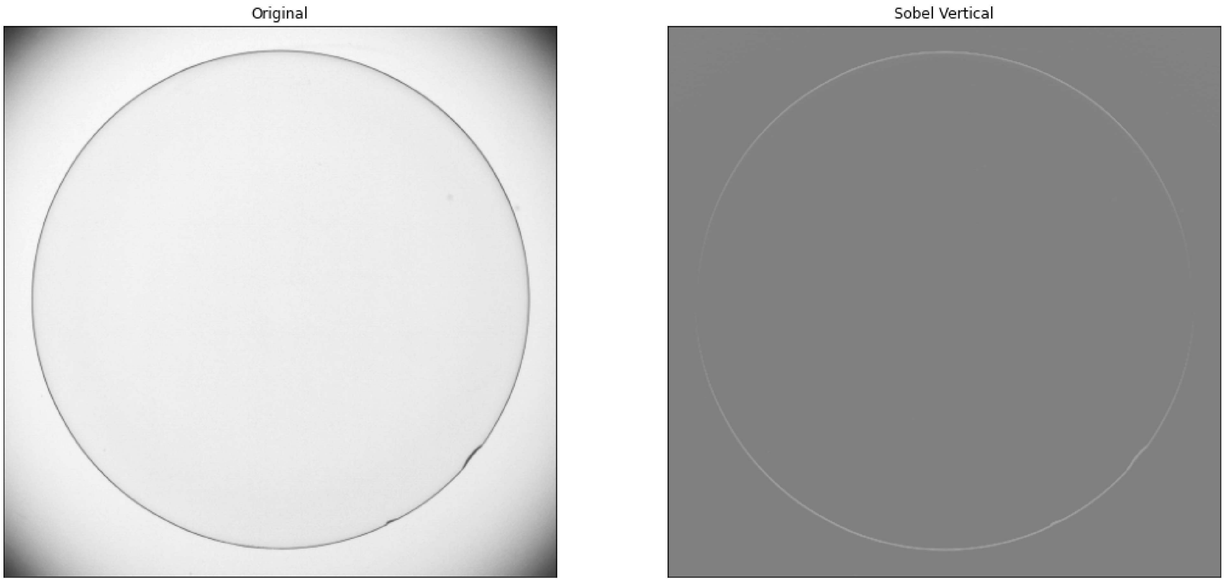
```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
<ipython-input-8-894218d2a2e0> in <module>
     27 ax.zaxis.set_major_locator(LinearLocator(10))
     28 ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))
---> 29 ax.set_aspect('equal', 'box')
     30 
     31 

~\AppData\Local\Programs\Python\Python39\lib\site-packages\mpl_toolkits\mplot3d\axes3
d.py in set_aspect(self, aspect, adjustable, anchor, share)
    321         """
    322         if aspect != 'auto':
--> 323             raise NotImplementedError(
    324                 "Axes3D currently only supports the aspect argument "
    325                 f"'auto'. You passed in {aspect!r}."

NotImplementedError: Axes3D currently only supports the aspect argument 'auto'. You p
assed in 'equal'.
```



In [ ]:

```
#Q3



img = cv.imread('contact_lens.tif', cv.IMREAD_REDUCED_GRAYSCALE_2)

# Sobel vertical
sobel_ver_kernel = np.array([(-1, -2, -1), (0, 0, 0), (1, 2, 1)], dtype='float32')
img_x = cv.filter2D(img,-1,sobel_ver_kernel)



fig, axes  = plt.subplots(1,2, sharex='all', sharey='all', figsize=(18,18))


axes[0].imshow(img, cmap='gray',vmin=0,vmax=255)
axes[0].set_title('Original')
axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(img_x, cmap='gray',vmin=-1020,vmax=1020)
axes[1].set_title('Sobel Vertical')
axes[1].set_xticks([]), axes[1].set_yticks([])
```

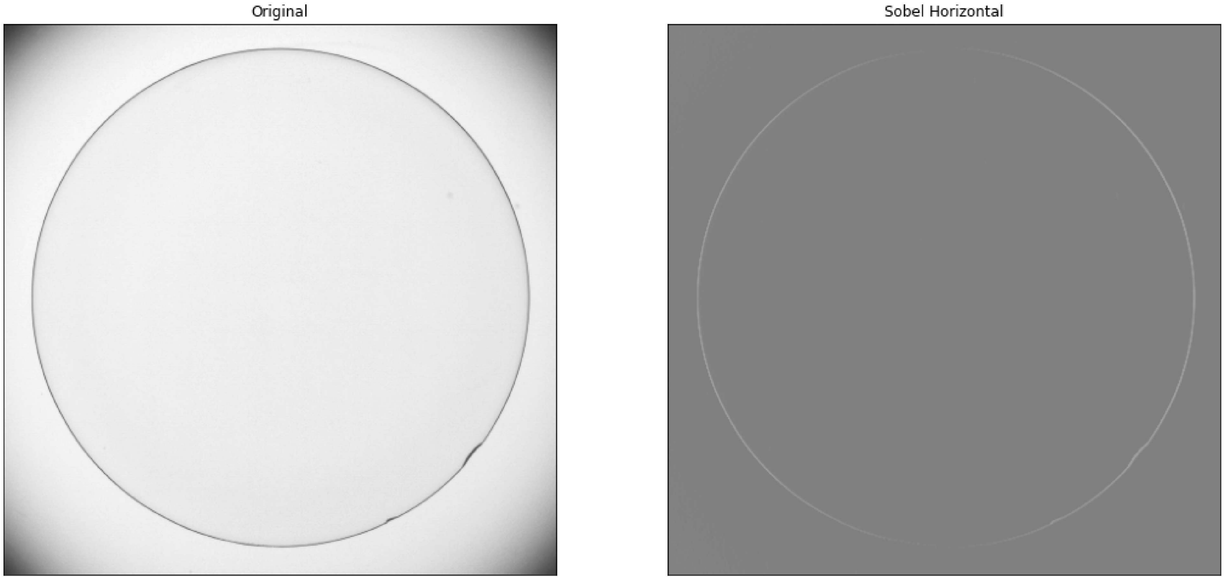
```
plt.show()
```


Original

Sobel Vertical

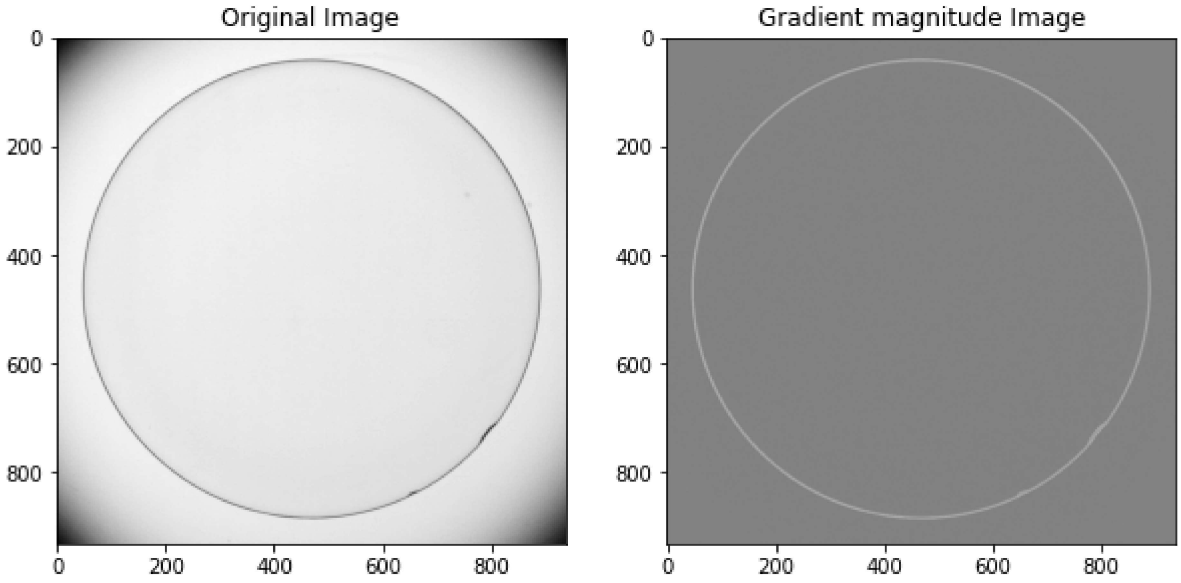
In [ ]:

```
# Sobel Horiontal


img = cv.imread('contact_lens.tif', cv.IMREAD_REDUCED_GRAYSCALE_2)


# Sobel horizontal
kernel = np.array([(-1, 0, 1), (-2, 0, 2), (-1, 0, 1)], dtype='float32')
img_y = cv.filter2D(img,-1,kernel)


fig, axes  = plt.subplots(1,2, sharex='all', sharey='all', figsize=(18,18))
axes[0].imshow(img, cmap='gray',vmin=0,vmax=255)
axes[0].set_title('Original')
axes[0].set_xticks([]), axes[0].set_yticks([])
axes[1].imshow(img_y, cmap='gray',vmin=-1020,vmax=1020)
axes[1].set_title('Sobel Horizontal')
axes[1].set_xticks([]), axes[1].set_yticks([])

plt.show()
```



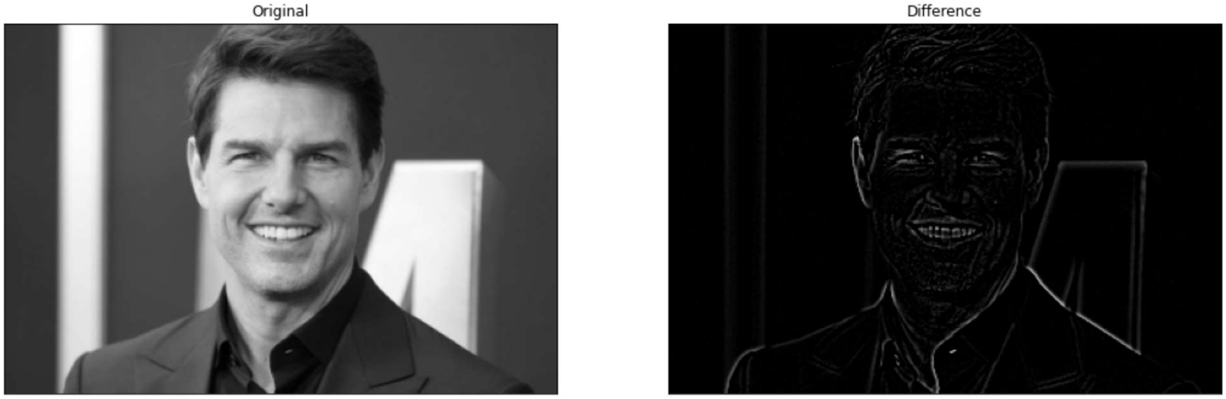
In [ ]:

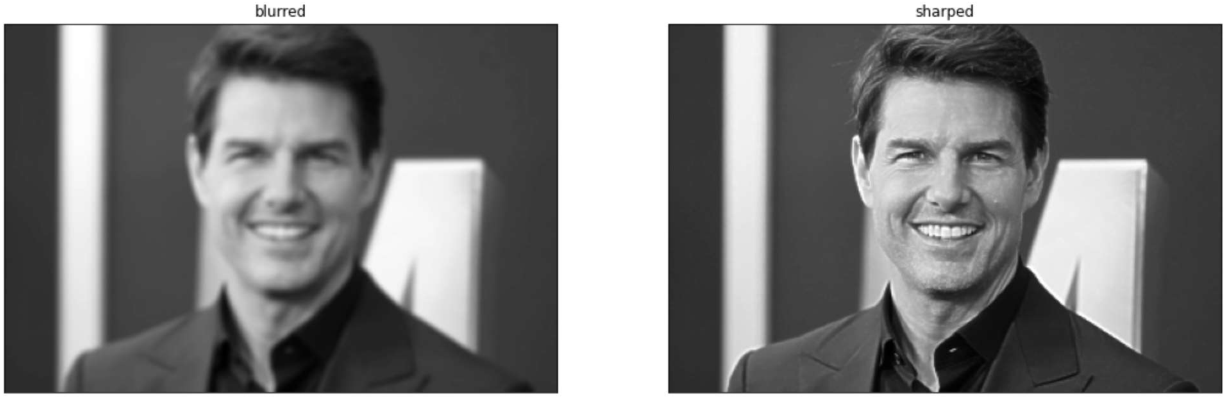
```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

img = cv.imread('contact_lens.tif', cv.IMREAD_GRAYSCALE).astype(np.float32)

Kernelx = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]])
Kernely = np.array([[1, 2, 1], [0, 0, 0], [-1, -2, -1]])

sobelx = cv.filter2D(img, -1, Kernelx)
sobely = cv.filter2D(img, -1, Kernely)

gm=np.sqrt(sobelx**2 +sobely**2)

fig, axes = plt.subplots(1,2, figsize=(10,10))
axes[0].imshow(img, cmap='gray')
axes[0].set_title('Original Image')
#axes[1].imshow(gm, cmap='gray')
axes[1].imshow(gm, cmap='gray',vmin=-1020,vmax=1020)

axes[1].set_title('Gradient magnitude Image')
plt.show()
```



In [ ]:

```
#Q4

tom = cv.imread('tom.jpg', cv.IMREAD_REDUCED_GRAYSCALE_2)

# Sobel vertical
sigma =2

gaussian_1D = cv.getGaussianKernel(5,sigma)
lp=cv.sepFilter2D(tom,-1,gaussian_1D,gaussian_1D,anchor=(-1,-1),delta=0,borderType=cv.
hp = cv.subtract(tom,lp)

sharp = cv.addWeighted(tom,1.0,hp,1.5,0)


fig, axes  = plt.subplots(2,2, sharex='all', sharey='all', figsize=(18,18))
axes[0][0].imshow(tom, cmap='gray')
axes[0][0].set_title('Original')
axes[0][0].set_xticks([]), axes[0][0].set_yticks([])
axes[1][0].imshow(lp, cmap='gray')
axes[1][0].set_title('blurred')
axes[1][0].set_xticks([]), axes[1][0].set_yticks([])
axes[0][1].imshow(hp, cmap='gray')
axes[0][1].set_title('Difference')
axes[0][1].set_xticks([]), axes[0][1].set_yticks([])
axes[1][1].imshow(sharp, cmap='gray')
axes[1][1].set_title('sharped')
axes[1][1].set_xticks([]), axes[1][1].set_yticks([])
plt.show()
```





```
In [ ]:
```