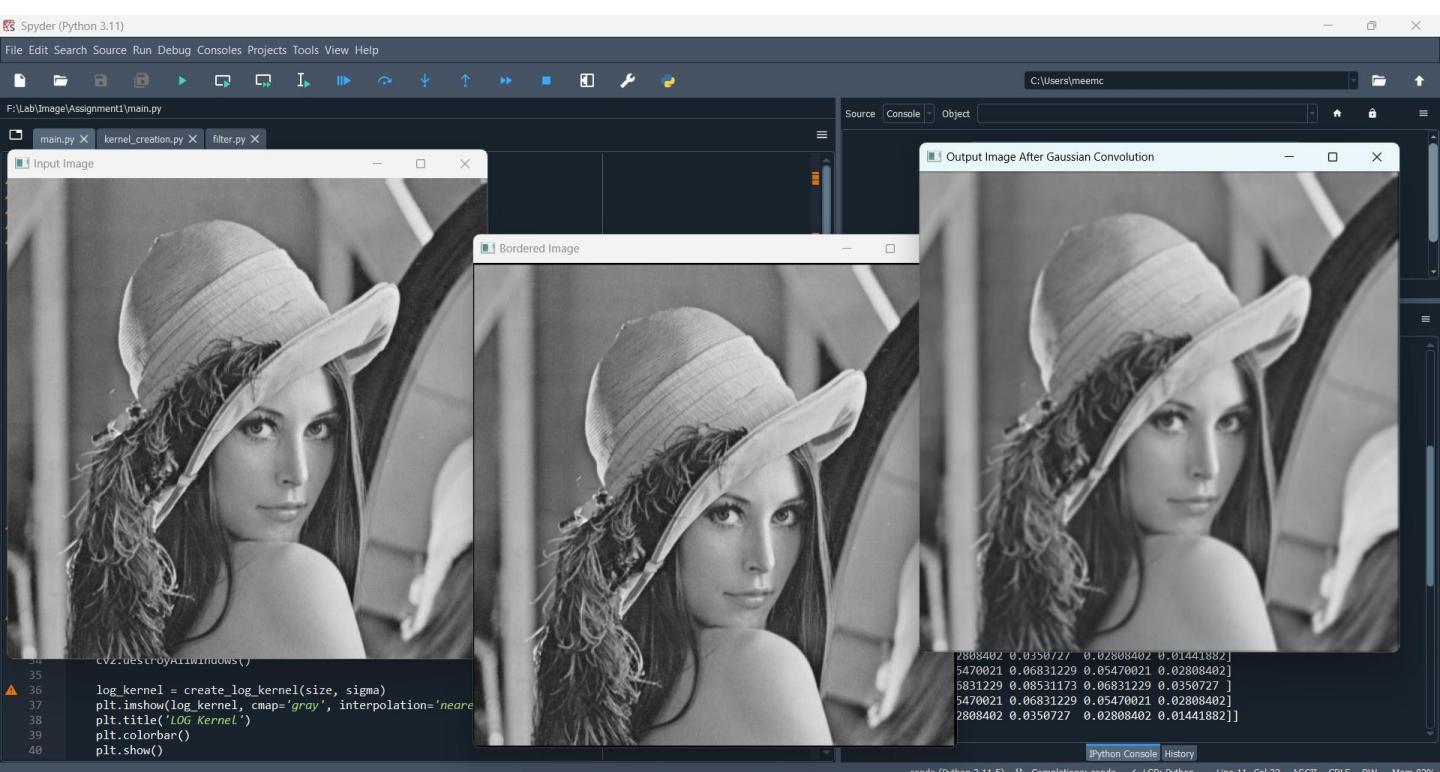
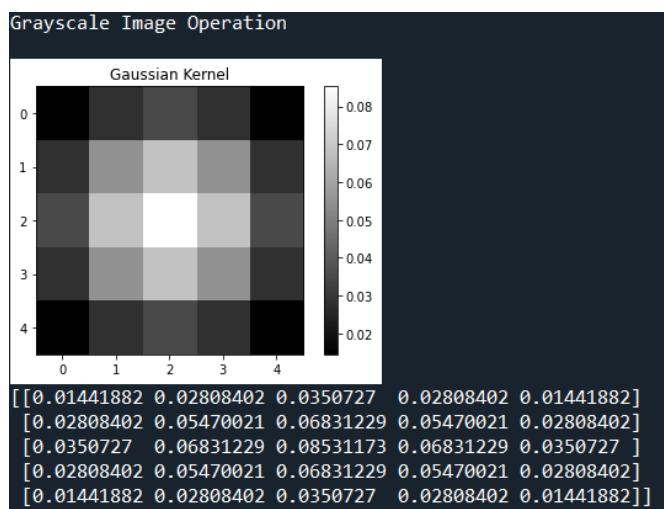
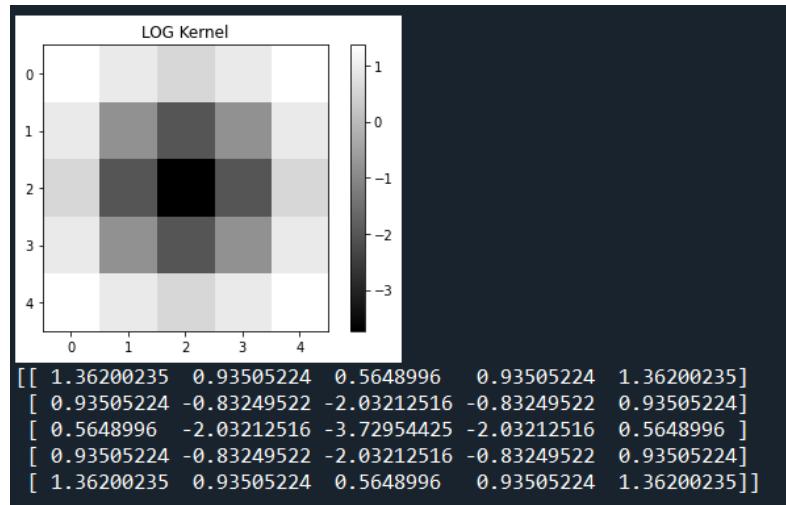
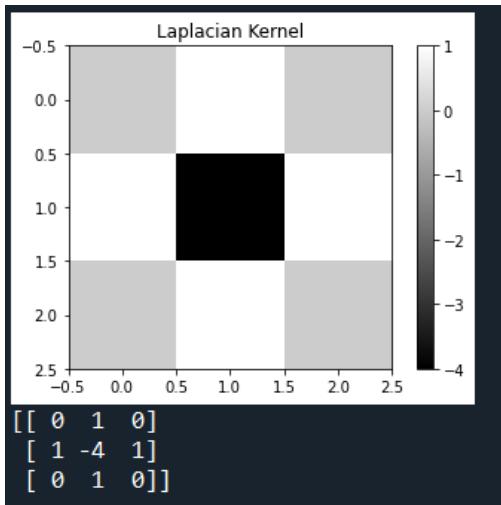


```
In [1]: runfile('F:/Lab/Image/Assignment1/main.py', wdir='F:/Lab/Image/Assignment1')
Choose Menu:
1. Grayscale Image
2. Color Image
3. HSV Image
4. Gaussian Subtraction of RGB & HSV
5. Different Filters
6. Exit
Enter your choice: 1
Enter Size of kernel: 5
Enter Sigma(SD) for kernel: 1.5
Enter kernel center x-coordinate: 2
Enter kernel center y-coordinate: 1
```





Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

F:\Lab\image\Assignment1\main.py

main.py X kernel\_creation.py X filter.py X

Input Image

Bordered Image

Output Image After Laplacian Convolution

Source Console Object

```
cv2.destroyAllWindows()
log_kernel = create_log_kernel(size, sigma)
plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
plt.title('LOG Kernel')
plt.colorbar()
plt.show()
```

Python Console History

Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

F:\Lab\image\Assignment1\main.py

main.py X kernel\_creation.py X filter.py X

Input Image

Bordered Image

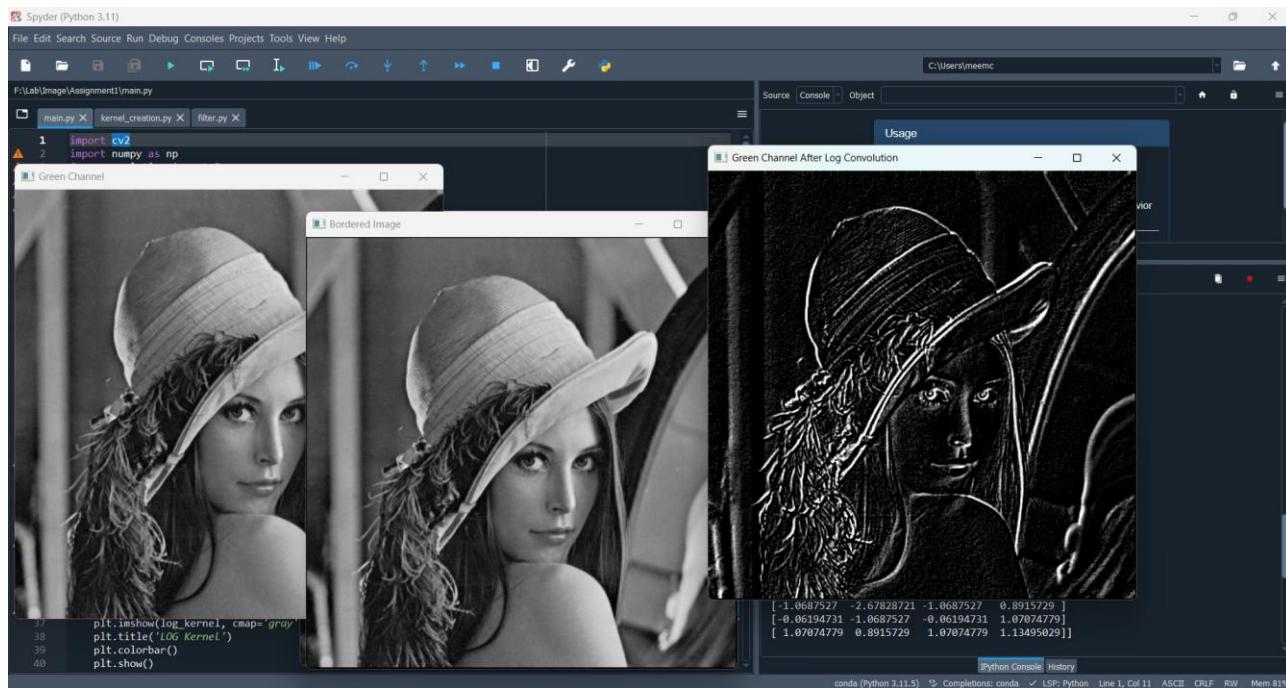
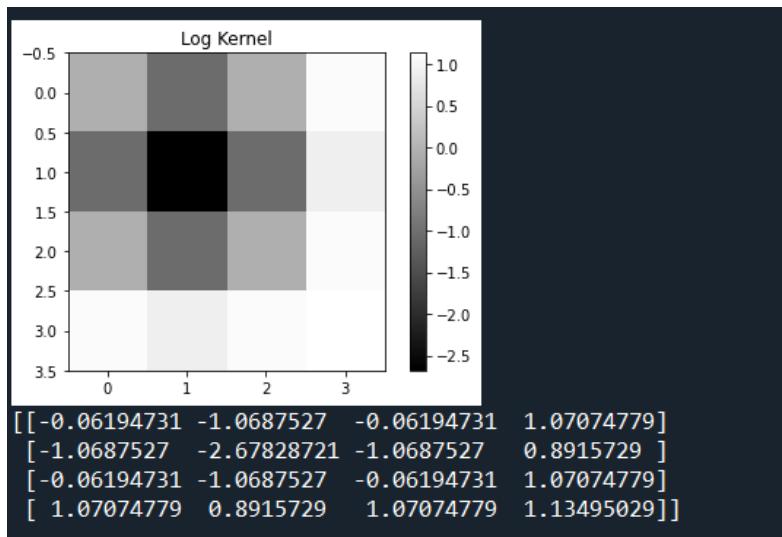
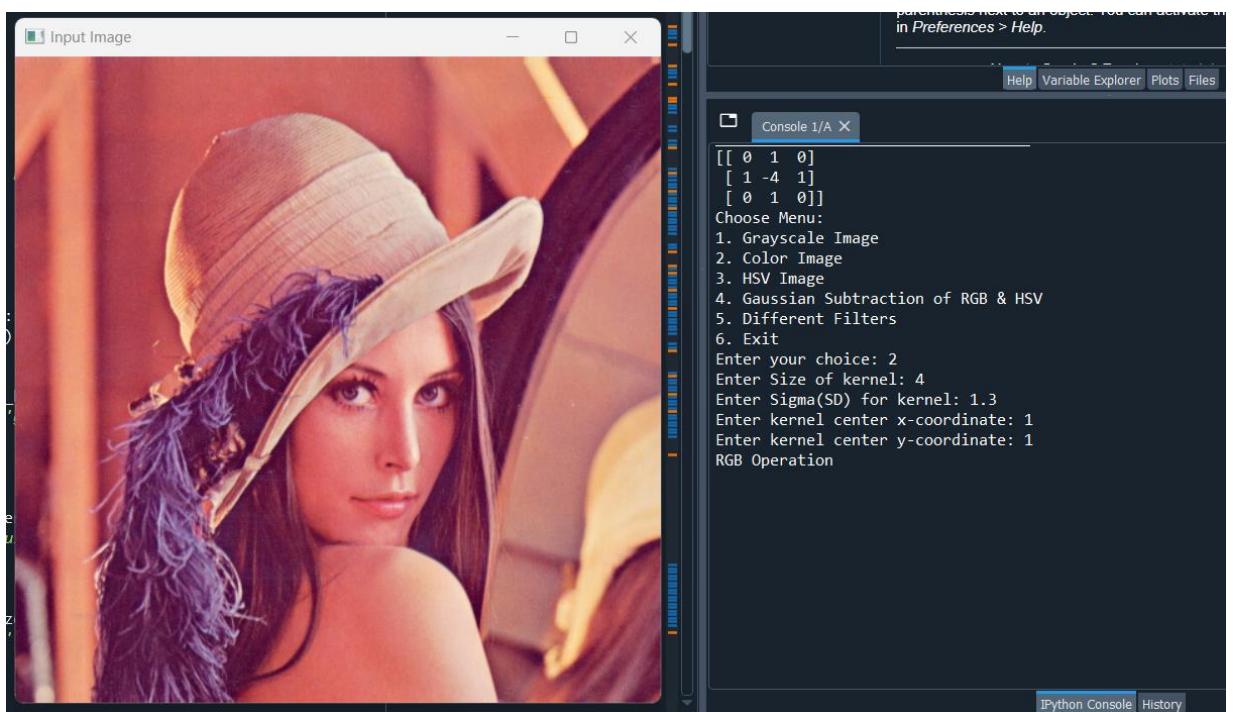
Output Image After LOG Convolution

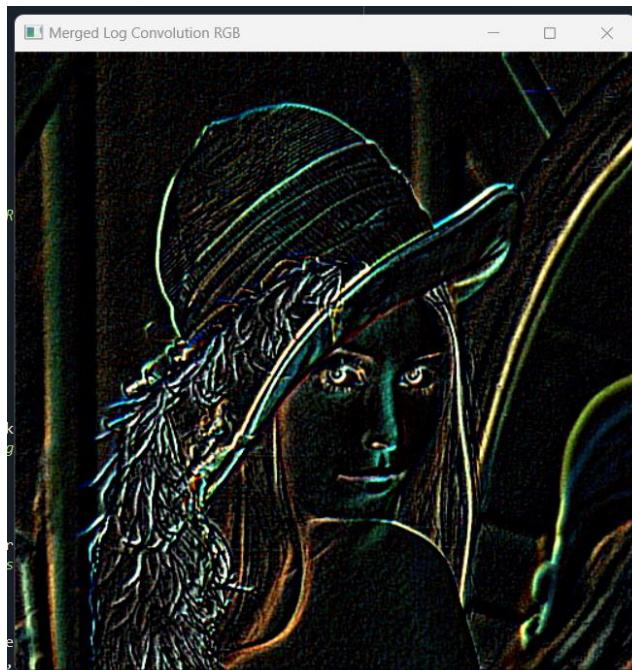
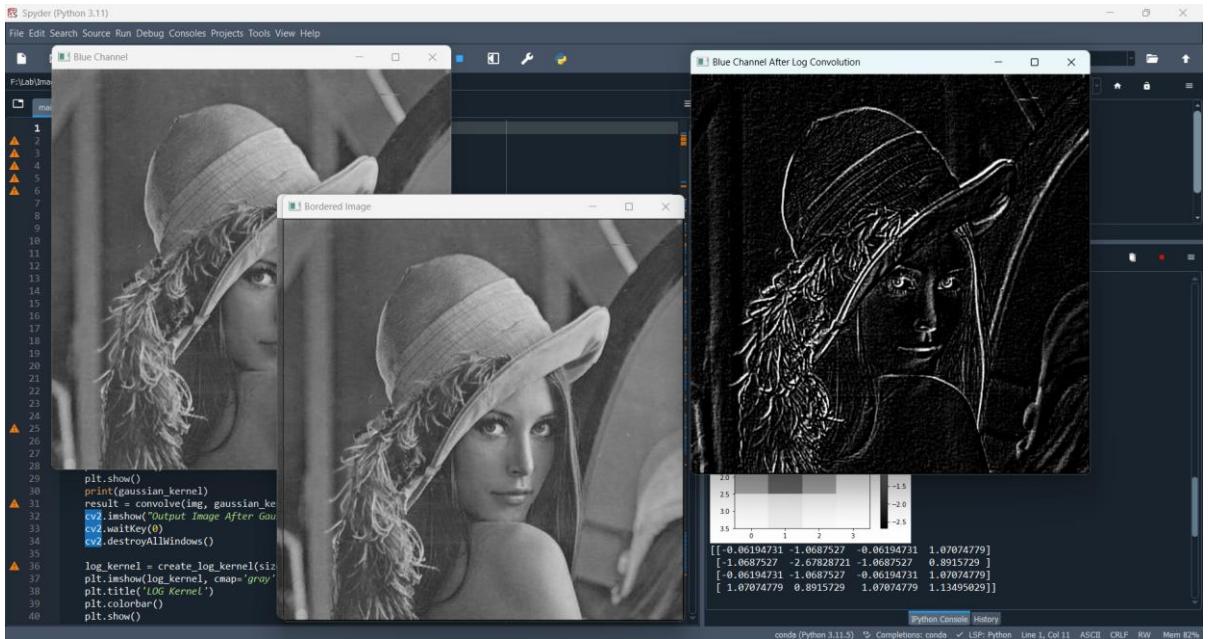
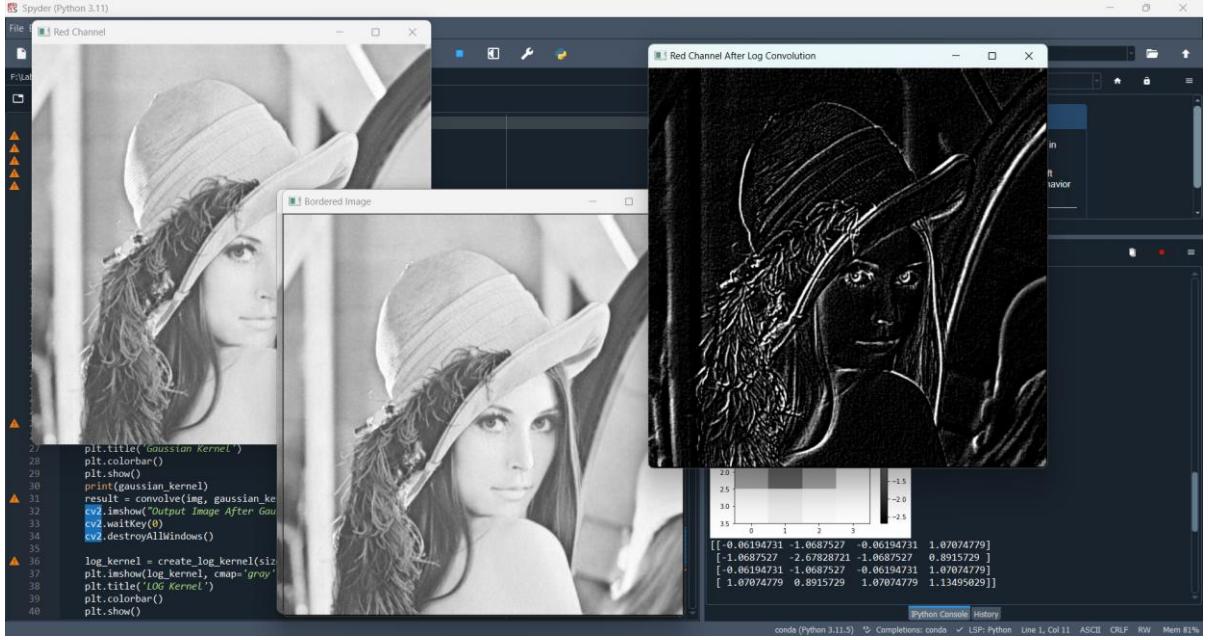
Source Console Object

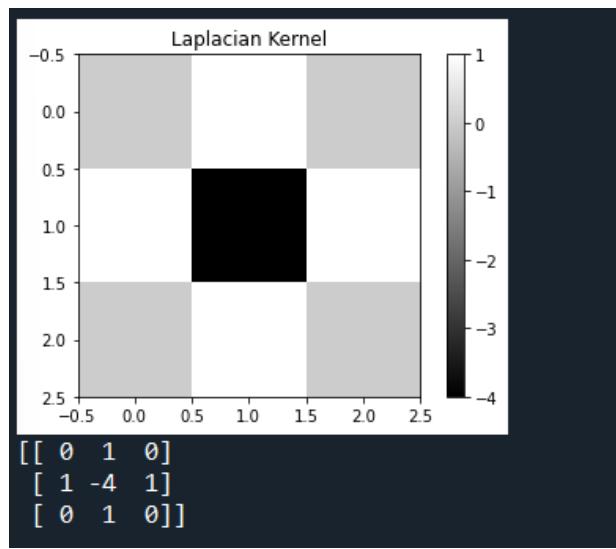
```
cv2.destroyAllWindows()
log_kernel = create_log_kernel(size, sigma)
plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
plt.title('LOG Kernel')
plt.colorbar()
plt.show()
```

Python Console History

```
[ 93505224  0.5648996   0.93505224  1.36200235]
[ 83249522 -2.03212516 -0.83249522  0.93505224]
[ 0.83249522 -3.72954425 -2.03212516  0.5648996 ]
[ 83249522 -2.03212516 -0.83249522  0.93505224]
[ 93505224  0.5648996   0.93505224  1.36200235]]
```







Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\meemc

main.py X kernel\_creation.py X filter.py X

```

1 import cv2
2 import numpy as np
3
4 # Green Channel
5 img = cv2.imread('lena.jpg')
6
7 # Bordered Image
8 log_kernel = np.array([[0, 1, 0], [1, -4, 1], [0, 1, 0]])
9
10 result = convolve(img, log_kernel)
11
12 cv2.imshow('Output Image After Log Convolution', result)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
16 gaussian_kernel = create_gaussian_kernel(3, 1)
17
18 result = convolve(img, gaussian_kernel)
19
20 cv2.imshow('Output Image After Gaussian Convolution', result)
21 cv2.waitKey(0)
22 cv2.destroyAllWindows()
23
24 log_kernel = create_log_kernel(3, 1)
25
26 result = convolve(img, log_kernel)
27
28 cv2.imshow('Output Image After Log Convolution', result)
29 cv2.waitKey(0)
30
31 print(gaussian_kernel)
32
33 result = convolve(img, gaussian_kernel)
34
35 cv2.imshow('Output Image After Gaussian Convolution', result)
36
37 log_kernel = create_log_kernel(3, 1)
38
39 result = convolve(img, log_kernel)
40
41 cv2.imshow('Output Image After Log Convolution', result)
42 cv2.waitKey(0)
43
44 cv2.destroyAllWindows()

```

Source Console Object

Usage

Green Channel After Laplacian Convolution

```

[[ 0  1  0]
 [ 1 -4  1]
 [ 0  1  0]]

```

Python Console History

conda (Python 3.11.5) ✓ Completions: conda ✓ LSP: Python Line 1, Col 11 ASCII CR LF RW

Spyder (Python 3.11)

File D Red Channel

C:\Users\meemc

main.py X kernel\_creation.py X filter.py X

```

1 import cv2
2 import numpy as np
3
4 # Red Channel
5 img = cv2.imread('lena.jpg')
6
7 # Bordered Image
8 log_kernel = np.array([[0, 1, 0], [1, -4, 1], [0, 1, 0]])
9
10 result = convolve(img, log_kernel)
11
12 cv2.imshow('Output Image After Log Convolution', result)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
16 gaussian_kernel = create_gaussian_kernel(3, 1)
17
18 result = convolve(img, gaussian_kernel)
19
20 cv2.imshow('Output Image After Gaussian Convolution', result)
21 cv2.waitKey(0)
22 cv2.destroyAllWindows()
23
24 log_kernel = create_log_kernel(3, 1)
25
26 result = convolve(img, log_kernel)
27
28 cv2.imshow('Output Image After Log Convolution', result)
29 cv2.waitKey(0)
30
31 print(gaussian_kernel)
32
33 result = convolve(img, gaussian_kernel)
34
35 cv2.imshow('Output Image After Gaussian Convolution', result)
36
37 log_kernel = create_log_kernel(3, 1)
38
39 result = convolve(img, log_kernel)
40
41 cv2.imshow('Output Image After Log Convolution', result)
42 cv2.waitKey(0)
43
44 cv2.destroyAllWindows()

```

Source Console Object

Usage

Red Channel After Laplacian Convolution

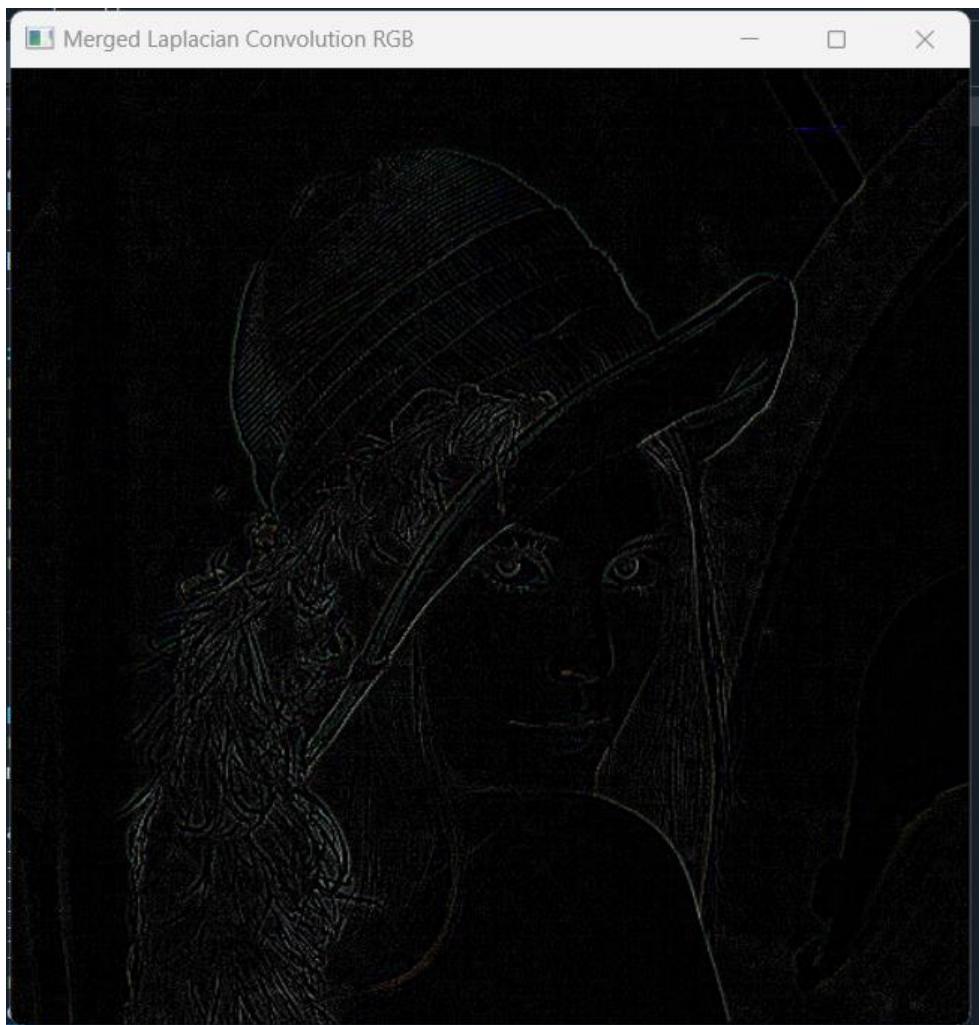
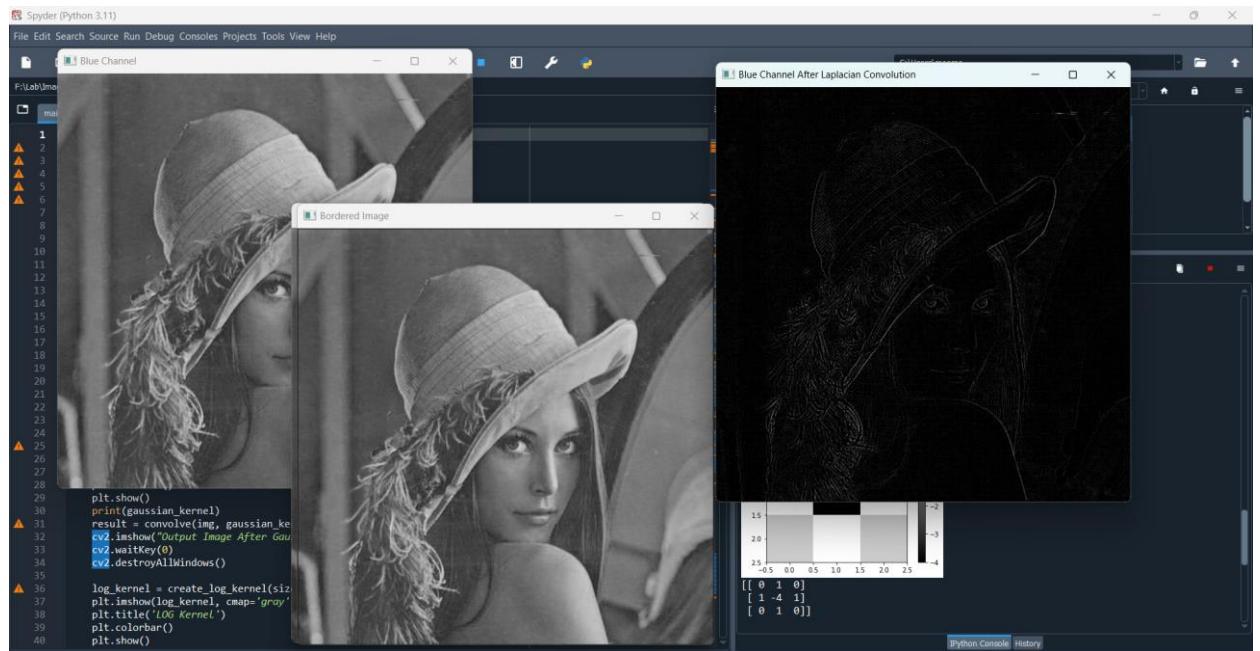
```

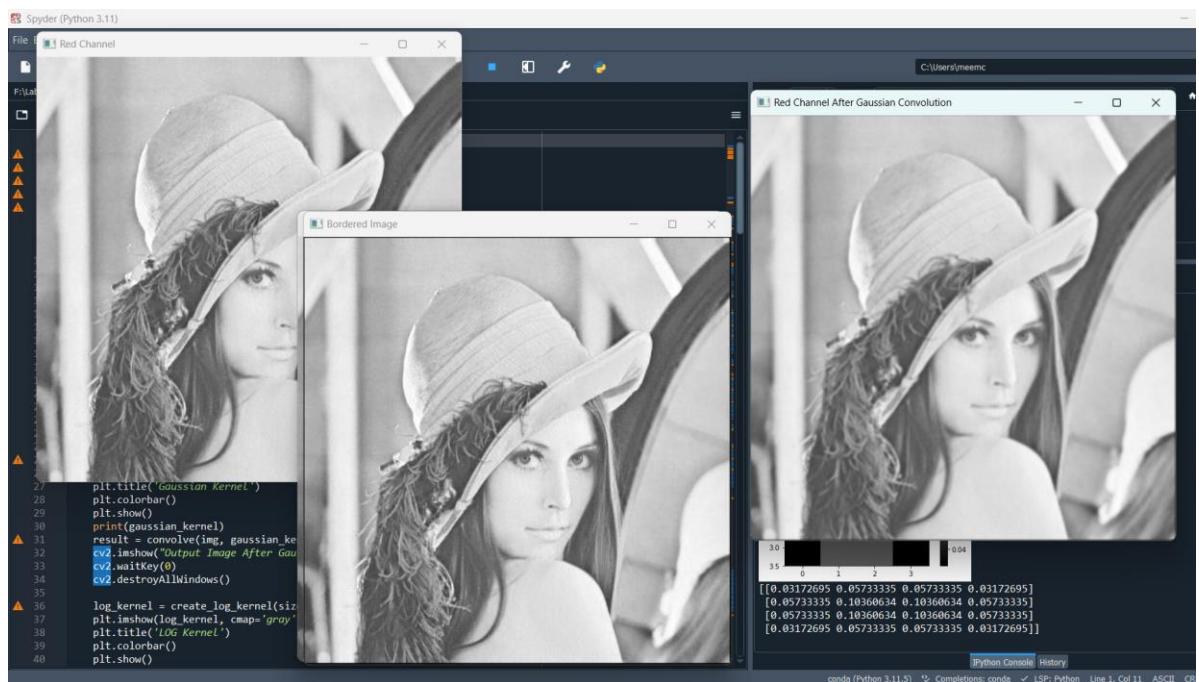
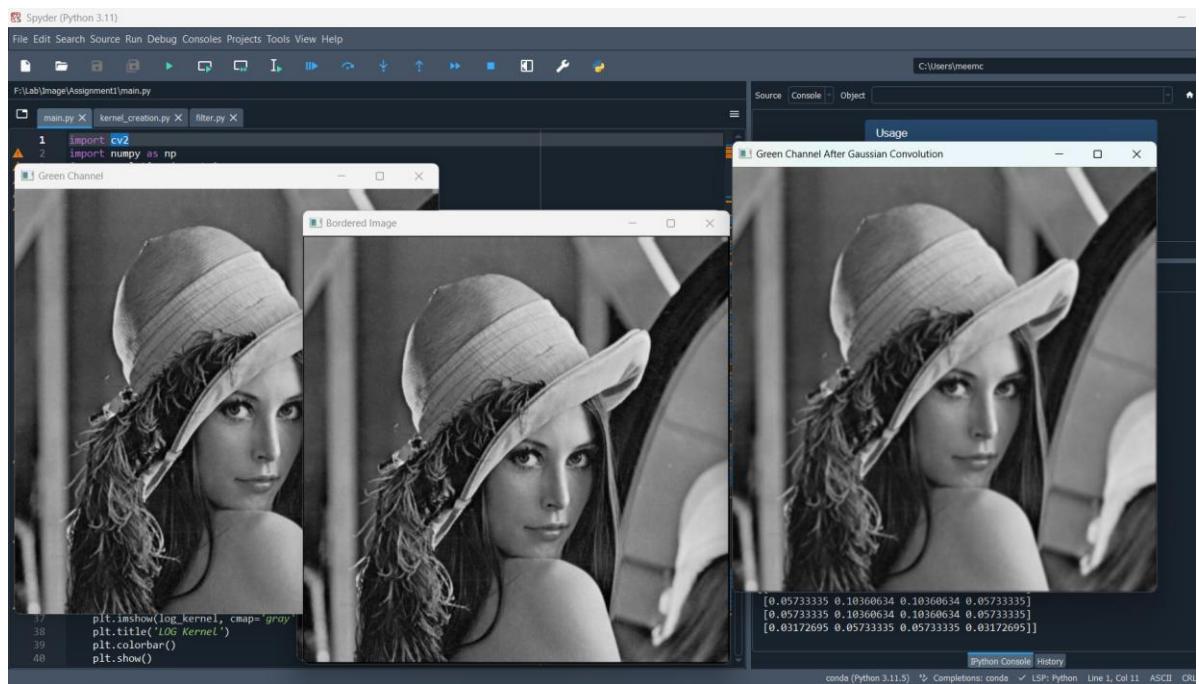
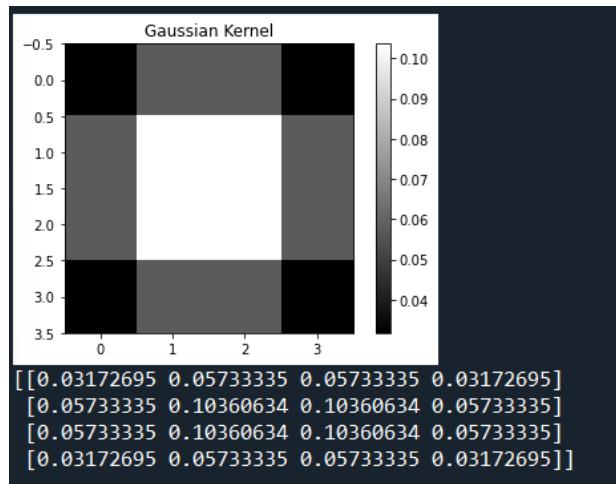
[[ 0  1  0]
 [ 1 -4  1]
 [ 0  1  0]]

```

Python Console History

conda (Python 3.11.5) ✓ Completions: conda ✓ LSP: Python Line 1, Col 11 ASCII CR LF RW





Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

F:\Lab\Ima... Blue Channel

ma...

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

plt.show()  
print(gaussian\_kernel)  
result = convolve(img, gaussian\_ke  
cv2.imshow("Output Image After Gau  
cv2.waitKey(0)  
cv2.destroyAllWindows()

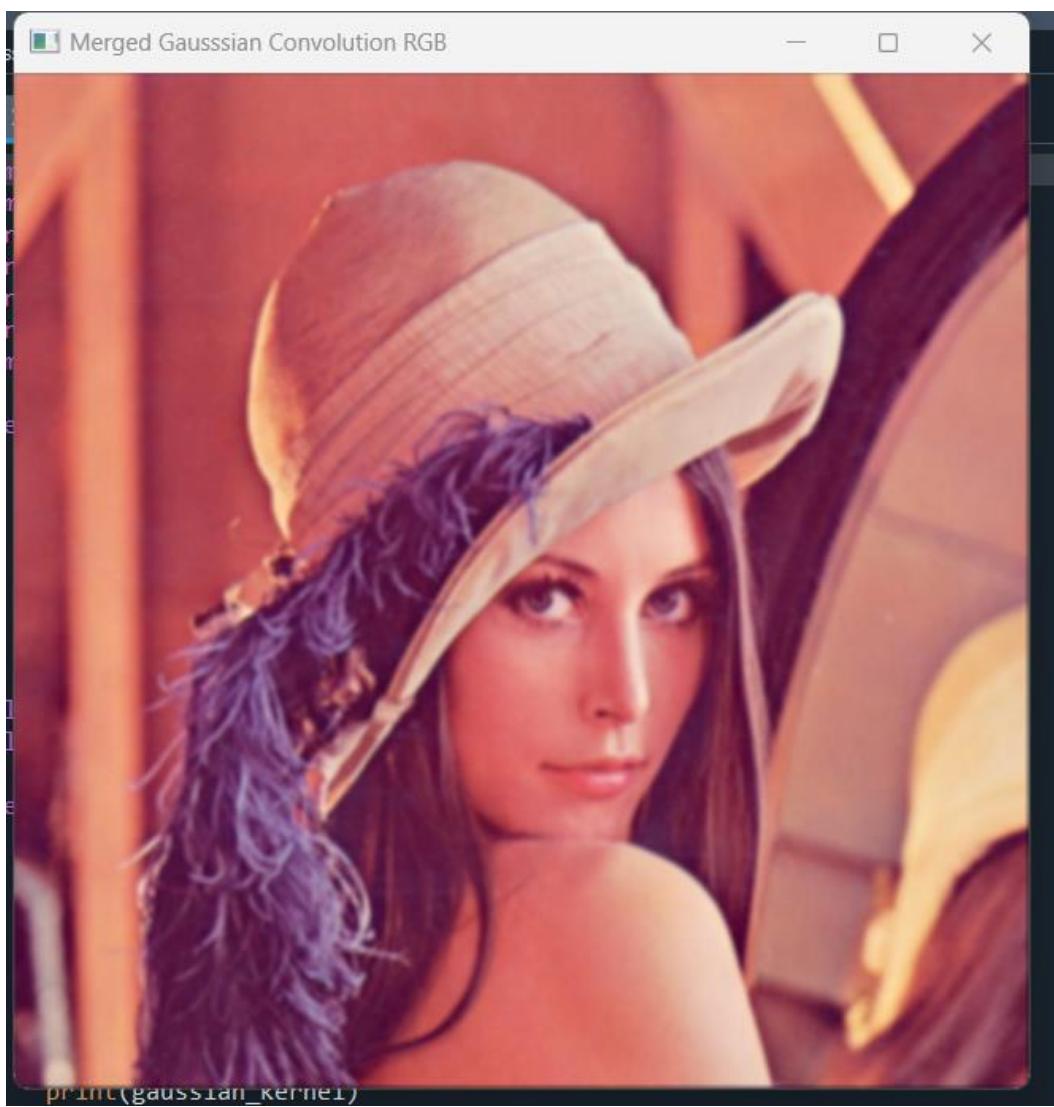
log\_kernel = create\_log\_kernel(siz  
plt.imshow(log\_kernel, cmap='gray'  
plt.title('LOG Kernel')  
plt.colorbar()  
plt.show()

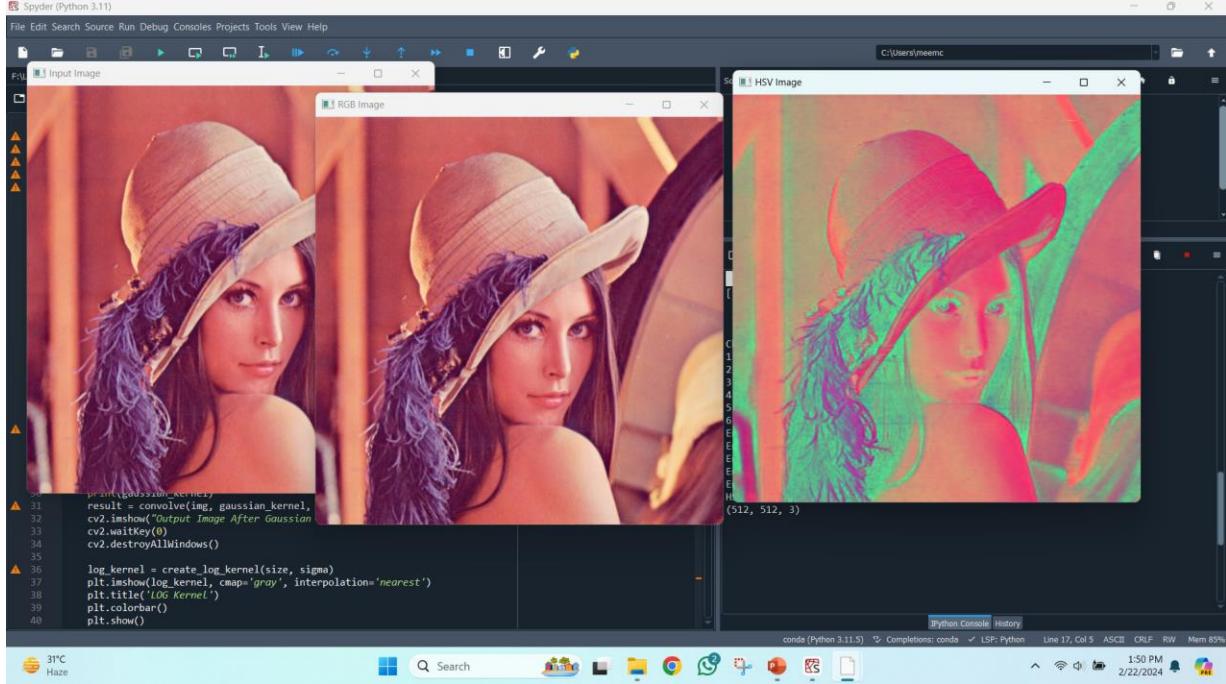
Blue Channel After Gaussian Convolution

Bordered Image

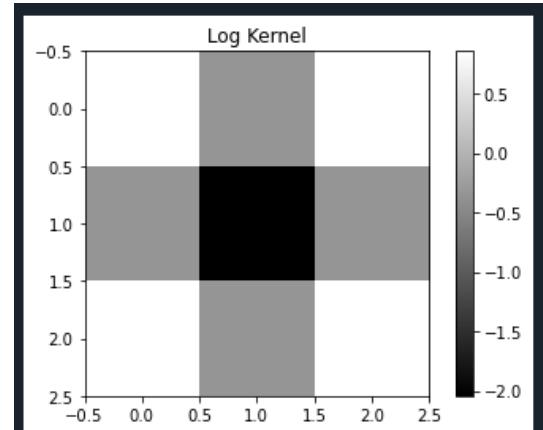
IPython Console History

conda (Python 3.11.5) Completions: conda LSP: Python Line 1, Col 11 ASCII CRLF RW





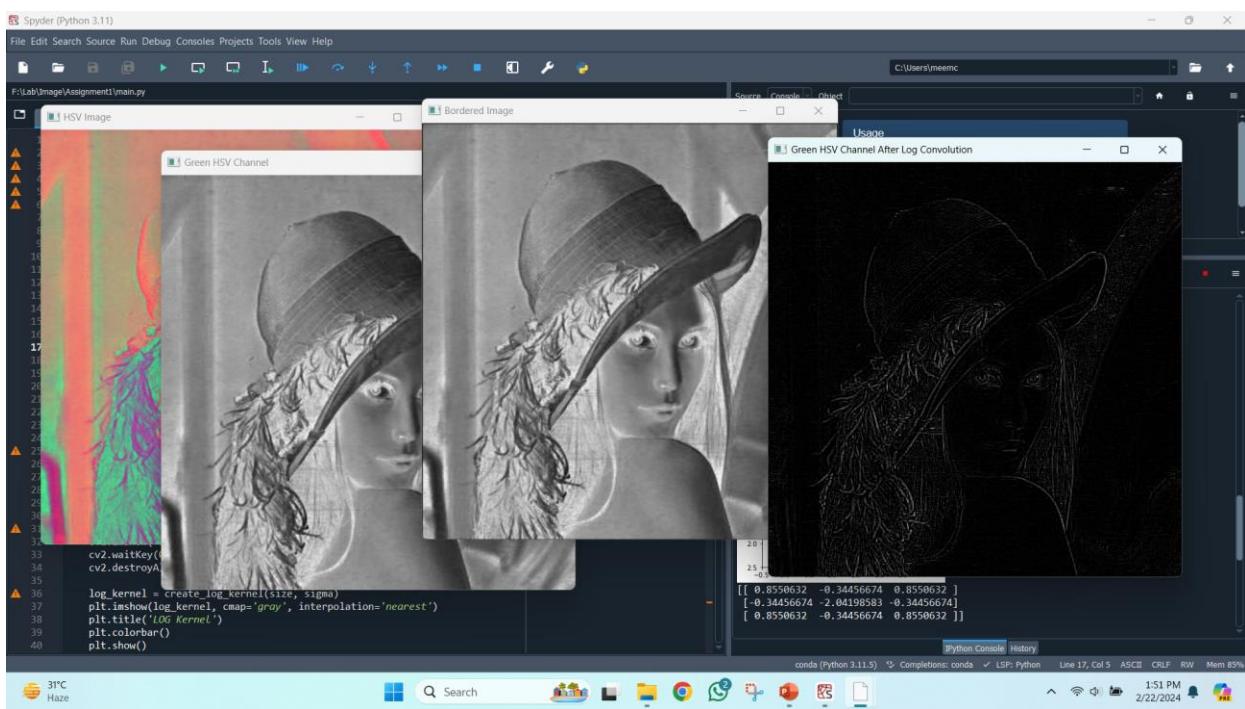
Choose Menu:  
 1. Grayscale Image  
 2. Color Image  
 3. HSV Image  
 4. Gaussian Subtraction of RGB & HSV  
 5. Different Filters  
 6. Exit  
 Enter your choice: 3  
 Enter Size of kernel: 3  
 Enter Sigma(SD) for kernel: 1.5  
 Enter kernel center x-coordinate: 1  
 Enter kernel center y-coordinate: 1  
 HSV Operation  
 (512, 512, 3)

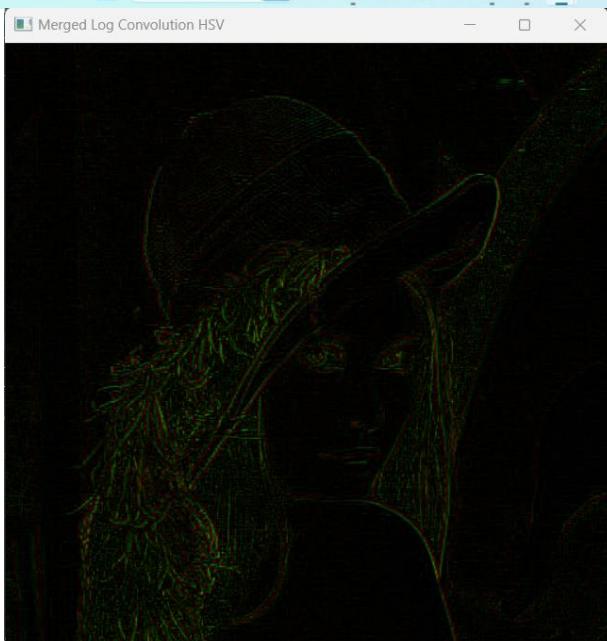
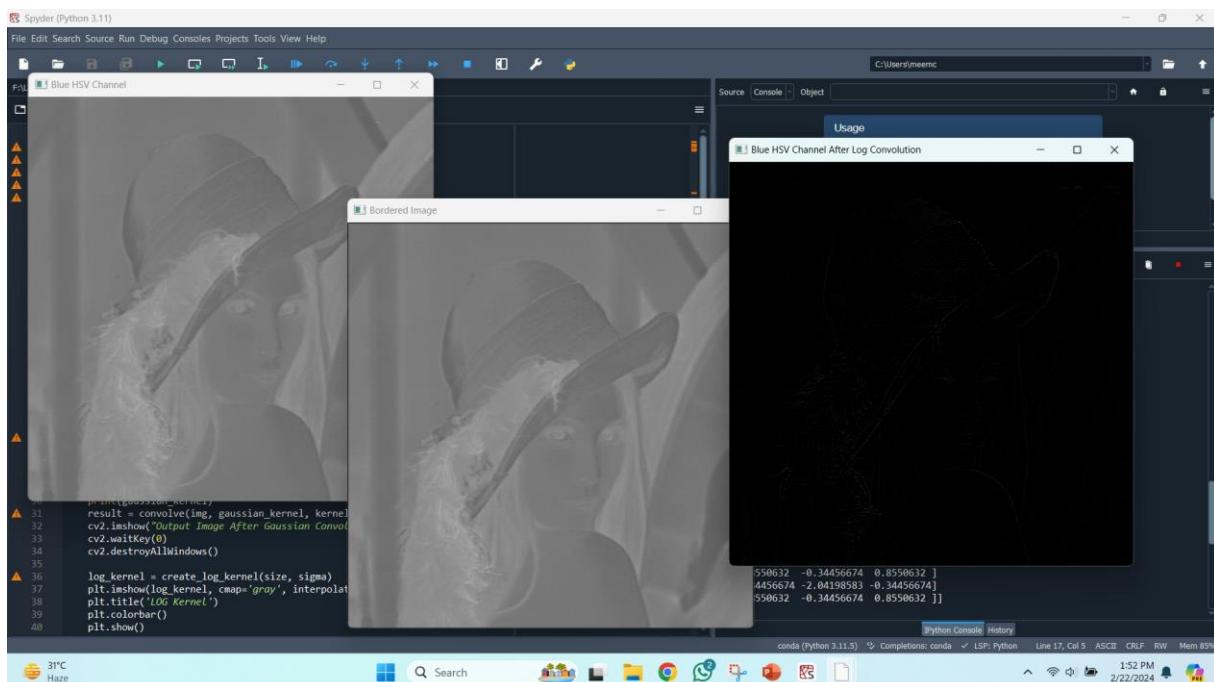
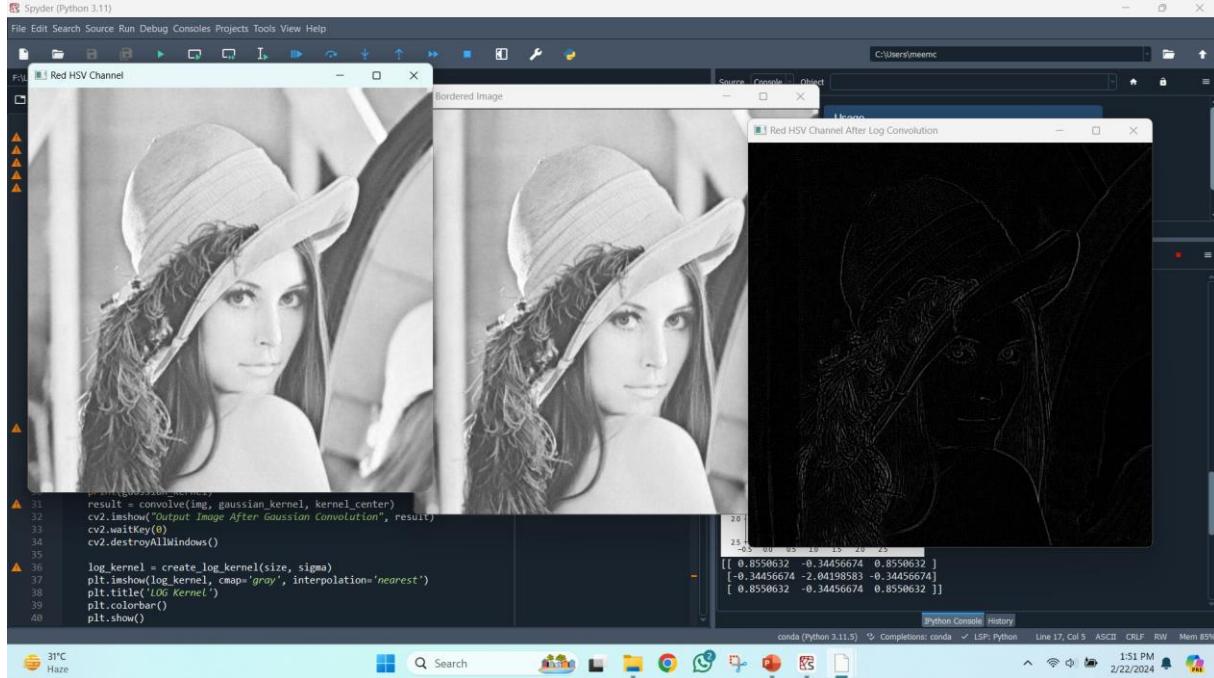


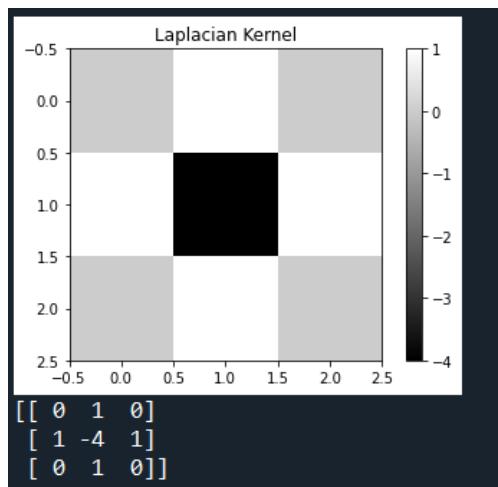
```

[[ 0.8550632 -0.34456674  0.8550632]
 [-0.34456674 -2.04198583 -0.34456674]
 [ 0.8550632 -0.34456674  0.8550632]]

```







Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

F:\Lab\Image\Assignment1\main.py

```

1  # Import required libraries
2  import cv2
3  import numpy as np
4  import matplotlib.pyplot as plt
5
6  # Load the input image
7  img = cv2.imread('lena.jpg')
8
9  # Create a bordered image
10 bordered_img = np.zeros((img.shape[0] + 2, img.shape[1] + 2, 3), dtype=np.uint8)
11 bordered_img[1:-1, 1:-1, :] = img
12
13 # Apply Laplacian convolution to the green channel
14 green_kernel = create_lap_kernel()
15 result = convolve(bordered_img[:, :, 1], green_kernel)
16 cv2.imshow("Green HSV Channel", result)
17 cv2.waitKey(0)
18 cv2.destroyAllWindows()
19
20 # Create a log kernel
21 log_kernel = create_log_kernel(size, sigma)
22 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
23 plt.title('LOG Kernel')
24 plt.colorbar()
25 plt.show()
26
27 # Apply Log convolution to the green channel
28 log_result = convolve(bordered_img[:, :, 1], log_kernel)
29 cv2.imshow("Bordered Image", log_result)
30 cv2.waitKey(0)
31 cv2.destroyAllWindows()
32
33 # Create a log kernel
34 log_kernel = create_log_kernel(size, sigma)
35 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
36 plt.title('LOG Kernel')
37 plt.colorbar()
38 plt.show()
39
40 # Create a log kernel
41 log_kernel = create_log_kernel(size, sigma)
42 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
43 plt.title('LOG Kernel')
44 plt.colorbar()
45 plt.show()

```

C:\Users\meemc

Source Console Object

Green HSV Channel After Laplacian Convolution

1 1  
1 0]

Python Console History

conda (Python 3.11.5) Completions: conda LSP: Python Line 17, Col 5 ASCII CRLF RW Mem 84%

31°C Haze

Search

1:58 PM 2/22/2024

Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

F:\Lab\Image\Assignment1\main.py

```

1  # Import required libraries
2  import cv2
3  import numpy as np
4  import matplotlib.pyplot as plt
5
6  # Load the input image
7  img = cv2.imread('lena.jpg')
8
9  # Create a bordered image
10 bordered_img = np.zeros((img.shape[0] + 2, img.shape[1] + 2, 3), dtype=np.uint8)
11 bordered_img[1:-1, 1:-1, :] = img
12
13 # Apply Gaussian convolution to the red channel
14 gaussian_kernel = create_gaussian_kernel()
15 result = convolve(bordered_img[:, :, 0], gaussian_kernel)
16 cv2.imshow("Input Image", result)
17 cv2.waitKey(0)
18 cv2.destroyAllWindows()
19
20 # Create a log kernel
21 log_kernel = create_log_kernel(size, sigma)
22 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
23 plt.title('LOG Kernel')
24 plt.colorbar()
25 plt.show()
26
27 # Apply Log convolution to the red channel
28 log_result = convolve(bordered_img[:, :, 0], log_kernel)
29 cv2.imshow("Red HSV Channel After Laplacian Convolution", log_result)
30 cv2.waitKey(0)
31 cv2.destroyAllWindows()
32
33 # Create a log kernel
34 log_kernel = create_log_kernel(size, sigma)
35 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
36 plt.title('LOG Kernel')
37 plt.colorbar()
38 plt.show()
39
40 # Create a log kernel
41 log_kernel = create_log_kernel(size, sigma)
42 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
43 plt.title('LOG Kernel')
44 plt.colorbar()
45 plt.show()

```

C:\Users\meemc

Source Console Object

Red HSV Channel After Laplacian Convolution

1 1  
1 0]

Python Console History

conda (Python 3.11.5) Completions: conda LSP: Python Line 17, Col 5 ASCII CRLF RW Mem 84%

31°C Haze

Search

1:58 PM 2/22/2024

Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Input Image

Bordered Image

C:\Users\meemc\Blue HSV Channel After Laplacian Convolution

```
30 import convolution as conv
31
32 result = convolve(img, gaussian_kernel, kernel)
33 cv2.imshow('Output Image After Gaussian Convolution', result)
34 cv2.waitKey(0)
35 cv2.destroyAllWindows()
36
37 log_kernel = create_log_kernel(size, sigma)
38 plt.imshow(log_kernel, cmap='gray', interpolation='none')
39 plt.title('LOG Kernel')
40 plt.colorbar()
41 plt.show()
```

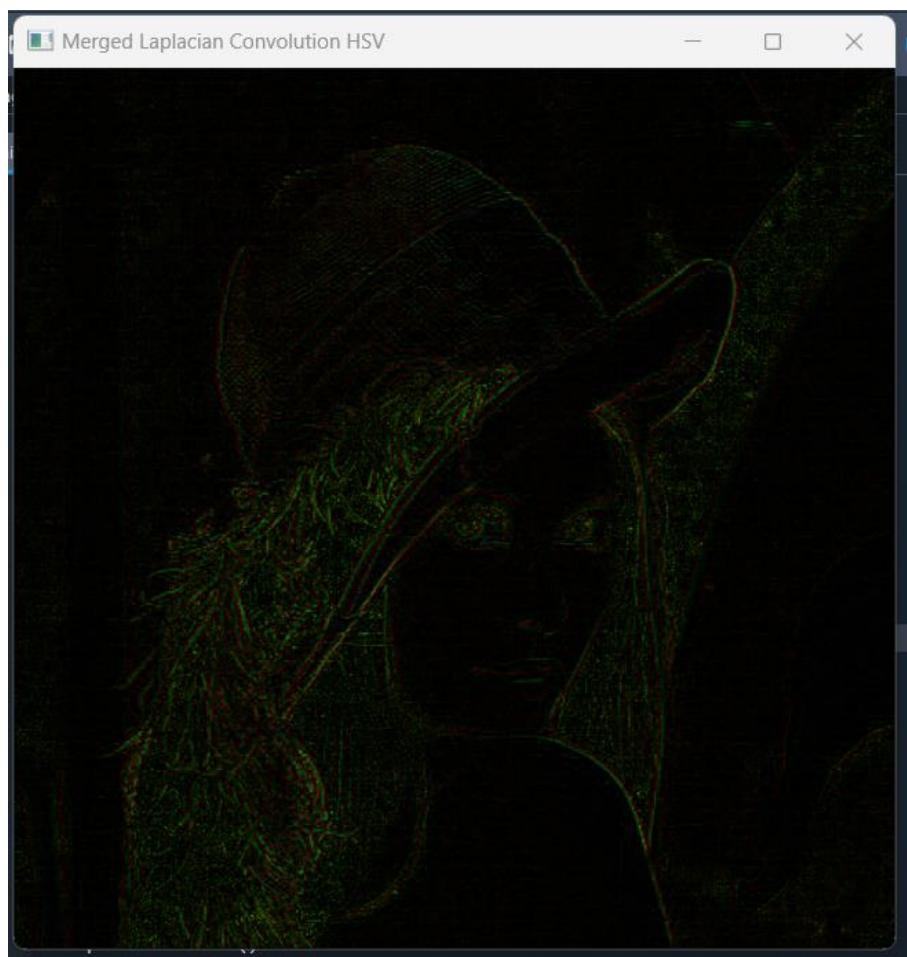
Python Console History

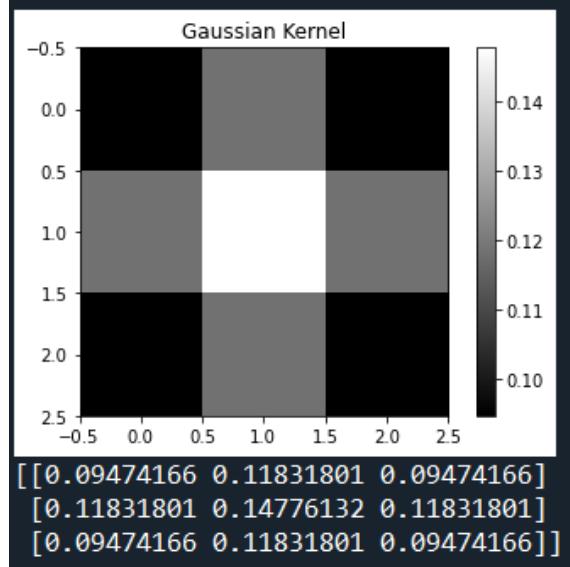
conda (Python 3.11.5) Completions: conda LSP: Python Line 17, Col 5 ASCII CRLF RW Mem 85%

31°C Haze

Search

1:58 PM 2/22/2024





The screenshot shows the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. The left sidebar shows project files: main.py, kernel\_creation.py, and filter.py. The main area displays three windows: 'Green HSV Channel' showing a grayscale image of a woman's face; 'Bordered Image' showing the same image with a thick black border; and 'Green HSV Channel After Gaussian Convolution' showing the image after convolution. Below these windows is a small color calibration bar. The bottom status bar shows system information: 31°C Haze, Python 3.11.5, LSP: Python, Line 17, Col 5, ASCII, CRLF, RW, Mem 85%. The bottom navigation bar includes icons for Home, Search, Google Drive, Folders, Google Sheets, Google Slides, Google Photos, and GitHub.

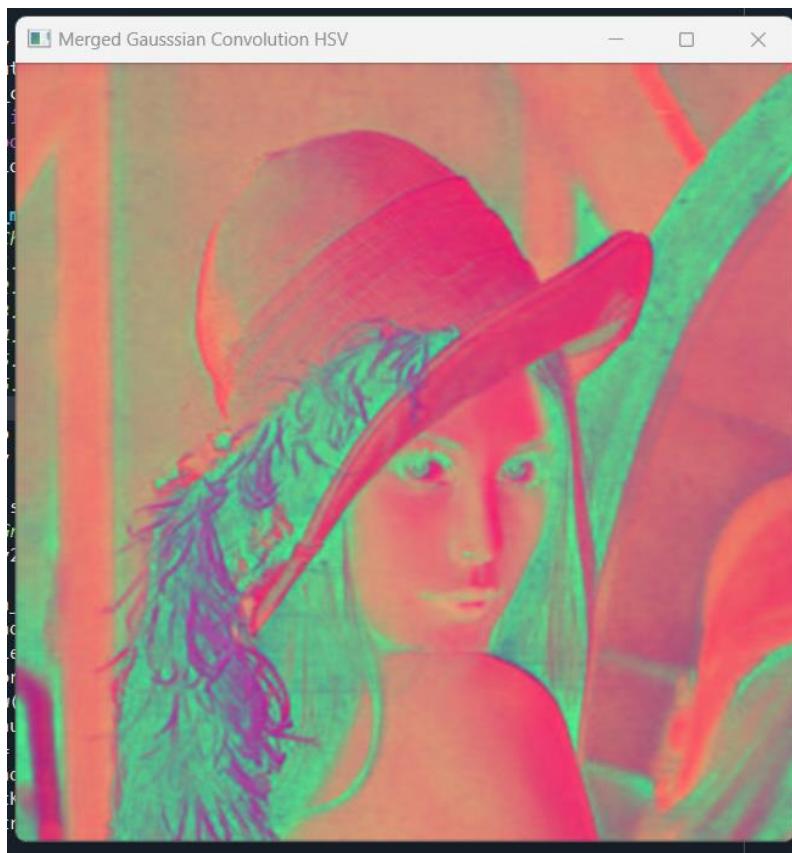
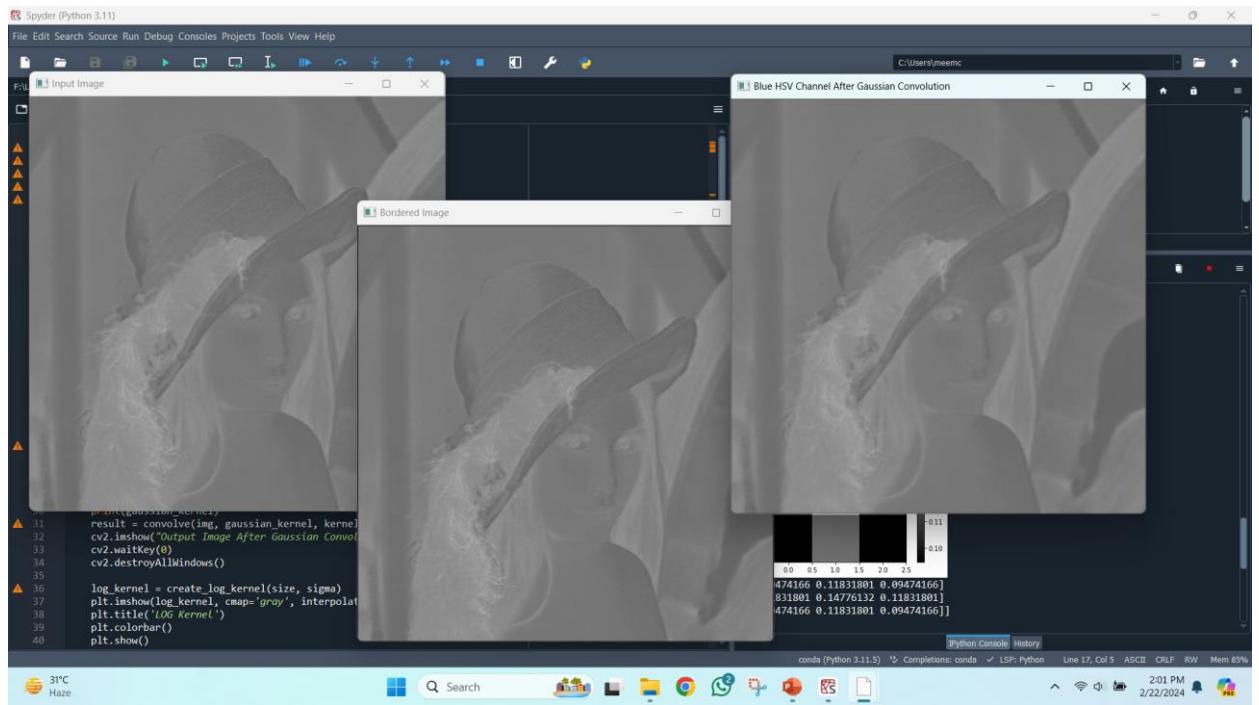
The screenshot shows the Spyder Python IDE interface with three windows and a console.

- Input Image:** A grayscale image of a woman wearing a hat.
- Bordered Image:** The same grayscale image with a thick black border around the central subject.
- Red HSV Channel After Gaussian Convolution:** A grayscale image showing the result of a convolution operation on the red channel of the HSV color space using a Gaussian kernel.
- Console:** Displays the following Python code and its output.

```
30 result = convolve(img, gaussian_kernel, kernel
31 cv2.imshow("Output Image After Gaussian Convolution", result)
32 cv2.waitKey(0)
33 cv2.destroyAllWindows()
34
35 log_kernel = create_log_kernel(size, sigma)
36 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
37 plt.title('Log Kernel')
38 plt.colorbar()
39 plt.show()
```

Output from the console:

Row	Column 1	Column 2	Column 3	Column 4	Column 5
0	0.00	0.10	0.20	0.30	0.40
1	0.474166	0.11831801	0.09474166		
2	0.831801	0.14776132	0.11831801		
3	0.474166	0.11831801	0.09474166		



Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

P:\Lab\image\Assignment1\main.py

main.py X Gaussian RGB Gaussian HSV Usage

```
1 import
2 import
3 from
4 from
5 from
6 from
7 import
8
9 def d:
10     p:
11     p:
12     p:
13     p:
14     p:
15     p:
16     p:
17     globa
18     globa
19
20 def op:
21     p:
22     i:
23
24 g:
25 p:
26 p:
27 p:
28 p:
29 p:
30 p:
31 r:
32 cv2.imshow("Output"
33 cv2.waitKey(0)
34 cv2.destroyAllWindows()
35
36 log_kernel = create_log_kernel(size, sigma)
37 plt.imshow(log_kernel, cmap='gray', interpolation='nearest')
38 plt.title('LOG Kernel')
39 plt.colorbar()
40 plt.show()
```

5. Different Filters  
6. Exit  
Enter your choice: 4

31°C Haze

Windows Taskbar: Search, Start, File Explorer, Edge, Google Chrome, WhatsApp, Telegram, File Manager, Paint, Task View, Taskbar Icons, Python Console, History, System Icons, Date/Time.

