

Kernels walk, **stride** and even jump all over images

Getting ready with IMPORTS

```
In [1]: import numpy as np
from scipy.linalg import orth
from scipy.ndimage import convolve
import skimage
from skimage import io
from skimage.exposure import histogram

# import plotting libraries
import plotly.express as px
import plotly.graph_objects as go

from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
%matplotlib inline
# also import the matrix printing function
!rm bug_numpy_utils.py
!wget https://raw.githubusercontent.com/bugrakoku/bug_python_utils/main/b
from bug_numpy_utils import MatPrint
from bug_numpy_utils import MatImshow

# download sample images
!rm *.jpg
!wget https://github.com/bugrakoku/data4all/raw/main/con1.jpg
!wget https://github.com/bugrakoku/data4all/raw/main/con2.jpg
!wget https://github.com/bugrakoku/data4all/raw/main/con3.jpg
!wget https://github.com/bugrakoku/data4all/raw/main/con4.jpg
!wget https://github.com/bugrakoku/data4all/raw/main/con5.jpg

print('done with importing necessary stuff')
```

```
--2024-12-15 15:43:20-- https://raw.githubusercontent.com/bugrakoku/bug_python_utils/main/bug_numpy_utils.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18456 (18K) [text/plain]
Saving to: 'bug_numpy_utils.py'
```

```
bug_numpy_utils.py 100%[=====>] 18.02K --KB/s in 0.009s
```

```
2024-12-15 15:43:20 (2.06 MB/s) - 'bug_numpy_utils.py' saved [18456/18456]
```

```
rm: cannot remove '*.jpg': No such file or directory
--2024-12-15 15:43:20-- https://github.com/bugrakoku/data4all/raw/main/con1.jpg
Resolving github.com (github.com)... 140.82.121.4
```

```
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/bugrakoku/data4all/main/con1.jpg [following]
```

```
--2024-12-15 15:43:21-- https://raw.githubusercontent.com/bugrakoku/data4all/main/con1.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.109.133, ...
```

```
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 245361 (240K) [image/jpeg]
Saving to: 'con1.jpg'
```

```
con1.jpg 100%[=====>] 239.61K 1.55MB/s in 0.2s
```

```
2024-12-15 15:43:22 (1.55 MB/s) - 'con1.jpg' saved [245361/245361]
```

```
--2024-12-15 15:43:22-- https://github.com/bugrakoku/data4all/raw/main/con2.jpg
Resolving github.com (github.com)... 140.82.121.4
```

```
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/bugrakoku/data4all/main/con2.jpg [following]
```

```
--2024-12-15 15:43:22-- https://raw.githubusercontent.com/bugrakoku/data4all/main/con2.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.111.133, 185.199.108.133, ...
```

```
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 232152 (227K) [image/jpeg]
Saving to: 'con2.jpg'
```

```
con2.jpg 100%[=====>] 226.71K 1.46MB/s in 0.2s
```

```
2024-12-15 15:43:23 (1.46 MB/s) - 'con2.jpg' saved [232152/232152]
```

```
--2024-12-15 15:43:23-- https://github.com/bugrakoku/data4all/raw/main/con3.jpg
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/bugrakoku/data4all/main/con3.jpg [following]
--2024-12-15 15:43:23-- https://raw.githubusercontent.com/bugrakoku/data4all/main/con3.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 288739 (282K) [image/jpeg]
Saving to: 'con3.jpg'
```

```
con3.jpg          100%[=====>] 281.97K  1.62MB/s    in 0.2s
```

2024-12-15 15:43:24 (1.62 MB/s) - 'con3.jpg' saved [288739/288739]

```
--2024-12-15 15:43:24-- https://github.com/bugrakoku/data4all/raw/main/con4.jpg
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/bugrakoku/data4all/main/con4.jpg [following]
--2024-12-15 15:43:25-- https://raw.githubusercontent.com/bugrakoku/data4all/main/con4.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 113548 (111K) [image/jpeg]
Saving to: 'con4.jpg'
```

```
con4.jpg          100%[=====>] 110.89K  --.-KB/s    in 0.1s
```

2024-12-15 15:43:25 (1.01 MB/s) - 'con4.jpg' saved [113548/113548]

```
--2024-12-15 15:43:25-- https://github.com/bugrakoku/data4all/raw/main/con5.jpg
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/bugrakoku/data4all/main/con5.jpg [following]
--2024-12-15 15:43:26-- https://raw.githubusercontent.com/bugrakoku/data4all/main/con5.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 299942 (293K) [image/jpeg]
Saving to: 'con5.jpg'
```

```
con5.jpg          100%[=====>] 292.91K  1.64MB/s   in 0.2
s
```

```
2024-12-15 15:43:26 (1.64 MB/s) - 'con5.jpg' saved [299942/299942]
```

```
done with importing necessary stuff
```

```
In [2]: !ls -al
```

```
total 5868
drwxrwxr-x 5 cago cago  4096 Dec 15 15:43 .
drwxrwxr-x 5 cago cago  4096 Dec 15 15:15 ..
drwxr-xr-x 2 cago cago  4096 Dec  6 20:51 AppleOrange
-rw-rw-r-- 1 cago cago 2938722 Dec 15 15:19 AppleOrange.zip
-rw-rw-r-- 1 cago cago  18456 Dec 15 15:43 bug_numpy_utils.py
-rw-rw-r-- 1 cago cago 245361 Dec 15 15:43 con1.jpg
-rw-rw-r-- 1 cago cago 232152 Dec 15 15:43 con2.jpg
-rw-rw-r-- 1 cago cago 288739 Dec 15 15:43 con3.jpg
-rw-rw-r-- 1 cago cago 113548 Dec 15 15:43 con4.jpg
-rw-rw-r-- 1 cago cago 299942 Dec 15 15:43 con5.jpg
-rw-rw-r-- 1 cago cago 1813991 Dec 15 15:40 ImagesKernels.ipynb
drwxrwxr-x 3 cago cago  4096 Dec 15 15:19 __MACOSX
-rw-rw-r-- 1 cago cago  3130 Dec 15 15:19 me536utils.py
drwxrwxr-x 2 cago cago  4096 Dec 15 15:43 __pycache__
-rw-rw-r-- 1 cago cago 16582 Dec 15 15:29 Week10_2738938.ipynb
```

```
# This is formatted as code
```

Verify convolution via Hadamard product

Just focus on the center of the data matrix

```
In [3]: a = np.array([[1,2,1], [1,5,1], [1,5,1]])
b = np.array( [[-1, -1, -1], [0,0,0], [1,1,1]] )
breflected = b[-1::-1,:]
MatPrint(a, 'data matrix')
MatPrint(b, 'b')
MatPrint(breflected, 'kernel')
print(f'\nsum of hadamard --> {(a*b).sum()}')
print(f'sum of hadamard with reflected b --> {(a*breflected).sum()}')
ab = convolve(a,b, mode='constant', cval=0)
MatPrint(ab, '\nconvolution of a.b')
print(f'\nObserve that {(a*breflected).sum()} = {ab[1,1]}\nmoving on...')
```

```

data matrix
|      1.00      2.00      1.00 |
|      1.00      5.00      1.00 |
|      1.00      5.00      1.00 |
b
|     -1.00     -1.00     -1.00 |
|      0.00      0.00      0.00 |
|      1.00      1.00      1.00 |
kernel
|      1.00      1.00      1.00 |
|      0.00      0.00      0.00 |
|     -1.00     -1.00     -1.00 |

sum of hadamard --> 3
sum of hadamard with reflected b --> -3

convolution of a.b
|     -6.00     -7.00     -6.00 |
|     -3.00     -3.00     -3.00 |
|      6.00      7.00      6.00 |

Observe that -3 = -3
moving on...

```

A function that does bunch of convolution over an image and a kernel

Zero-, one-padding and reflection alternatives are implemented

```

In [4]: def ConvolveAlternatives(I, K):
        print(f'\nOriginal Image is {I.shape[0]} by {I.shape[1]}')
        MatImshow(I, title='original image')
        MatPrint(K, 'Kernel:')
        MatImshow(convolve(I,B,mode='reflect'), 'reflect')
        MatImshow(convolve(I,B,mode='constant', cval=0), 'zero padding')
        MatImshow(convolve(I,B,mode='constant', cval=1), 'one padding')
        MatImshow(convolve(I,B,mode='nearest'), 'nearest')
        MatImshow(convolve(I,B,mode='mirror'), 'mirror')
        MatImshow(convolve(I,B,mode='wrap'), 'wrap')

```

Generate Test Image

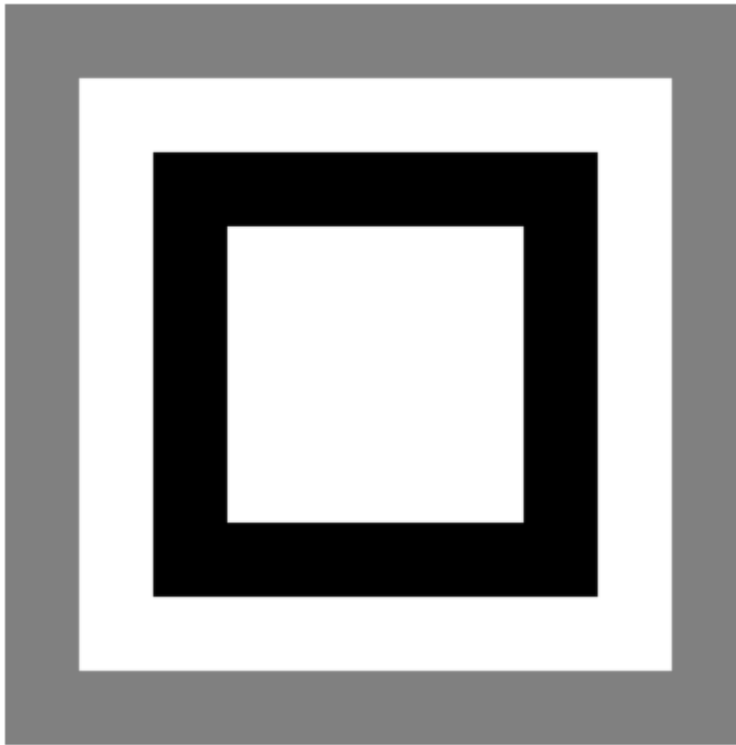
```

In [5]: I = np.zeros((10,10))
        I[(2,7),2:8] = 1
        I[3:8,(2,7)] = 1
        I[(0,9),:] = 0.5
        I[:,(0,9)] = 0.5
        I = 1 - I
        print(f'Original Image is {I.shape[0]} by {I.shape[1]}')
        dummy = MatImshow(I, title='Original Image')

```

Original Image is 10 by 10

Original Image



Simple averaging kernels

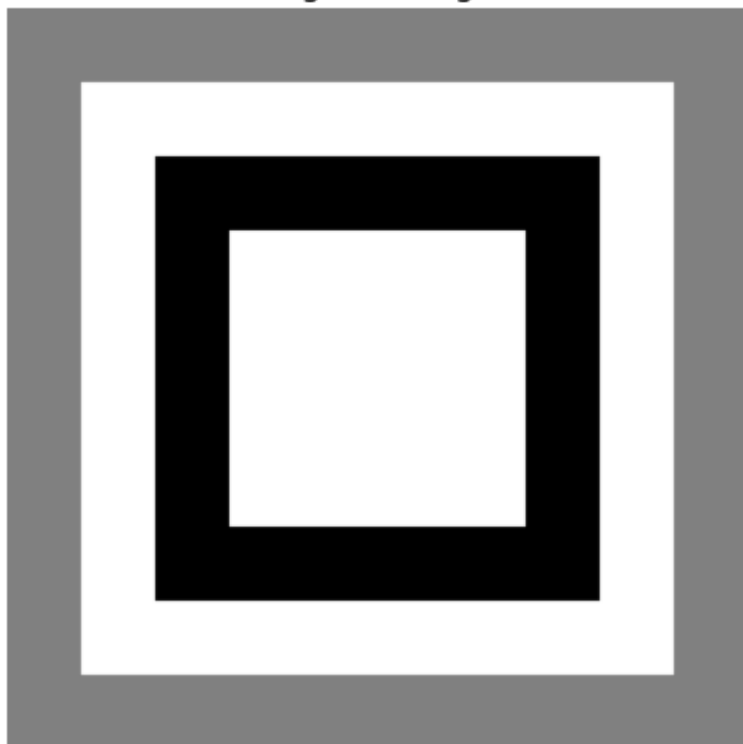
3×3 and 5×5 will be tested

```
In [6]: # simple averaging Kernel
print(f'\n\n3x3 averaging kernel')
B = np.ones((3,3))/9
ConvolveAlternatives(I,B)
```

3x3 averaging kernel

Original Image is 10 by 10

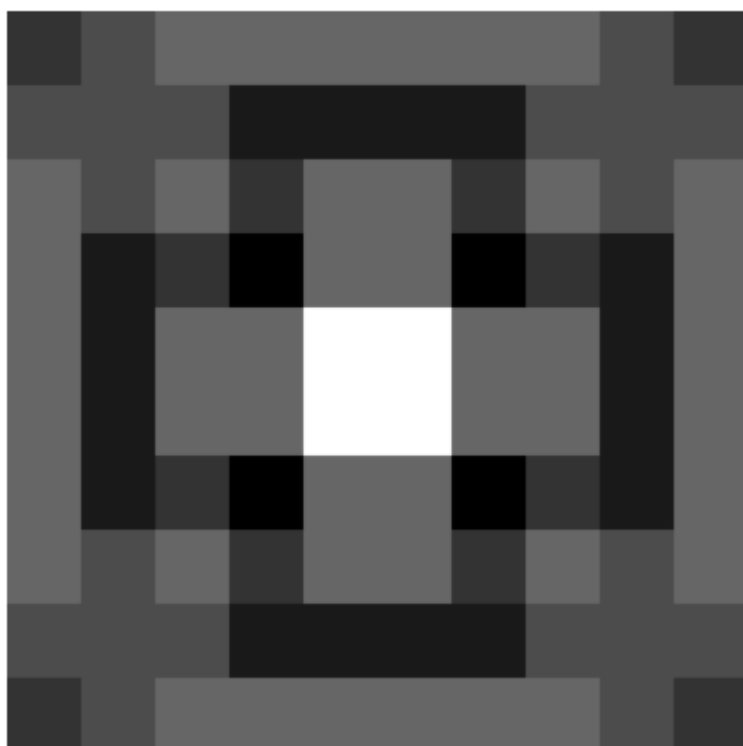
original image



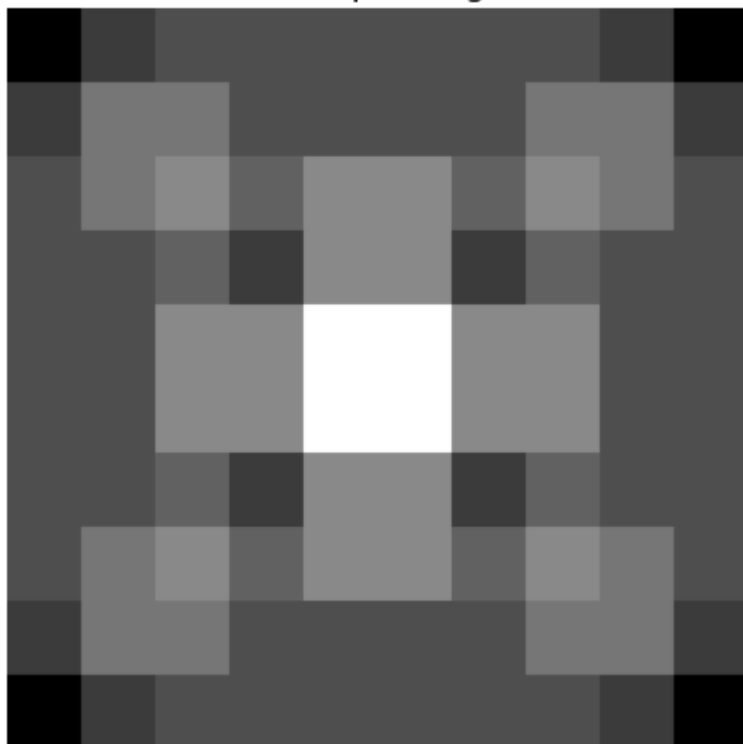
Kernel:

	0.11	0.11	0.11	
	0.11	0.11	0.11	
	0.11	0.11	0.11	

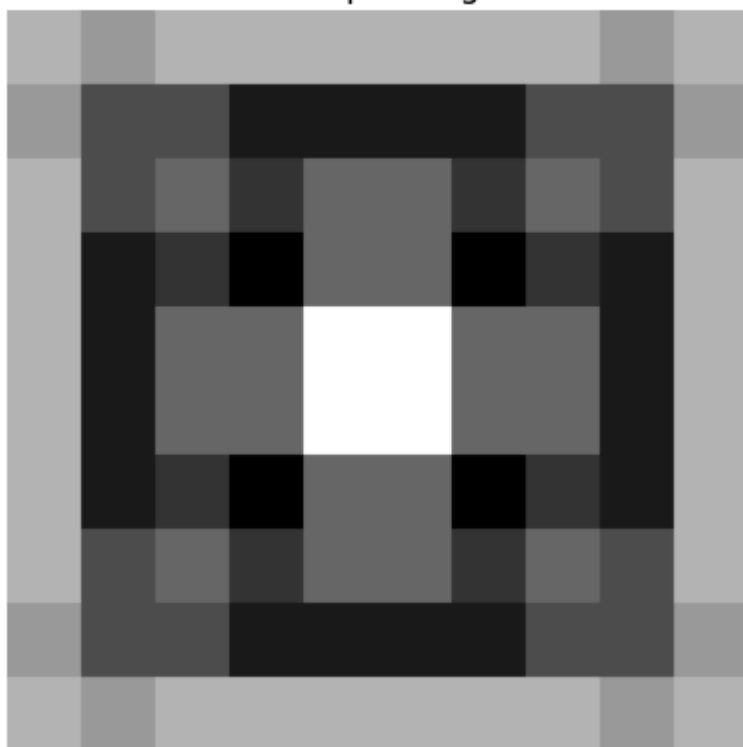
reflect



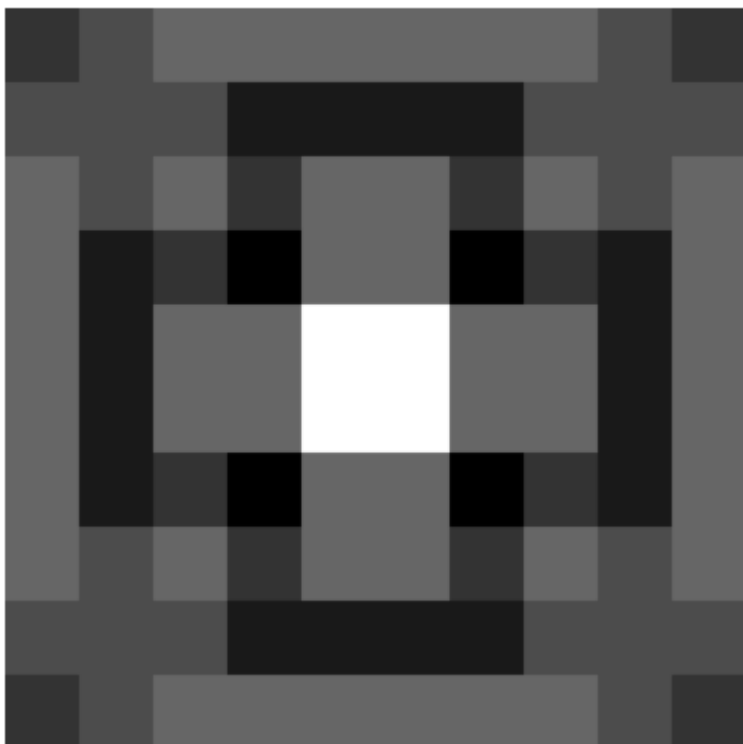
zero padding



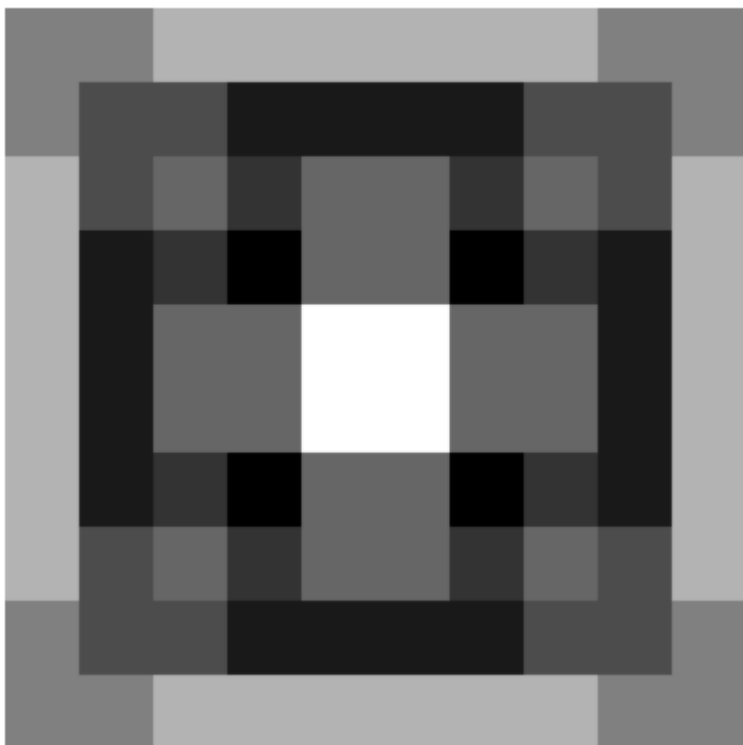
one padding



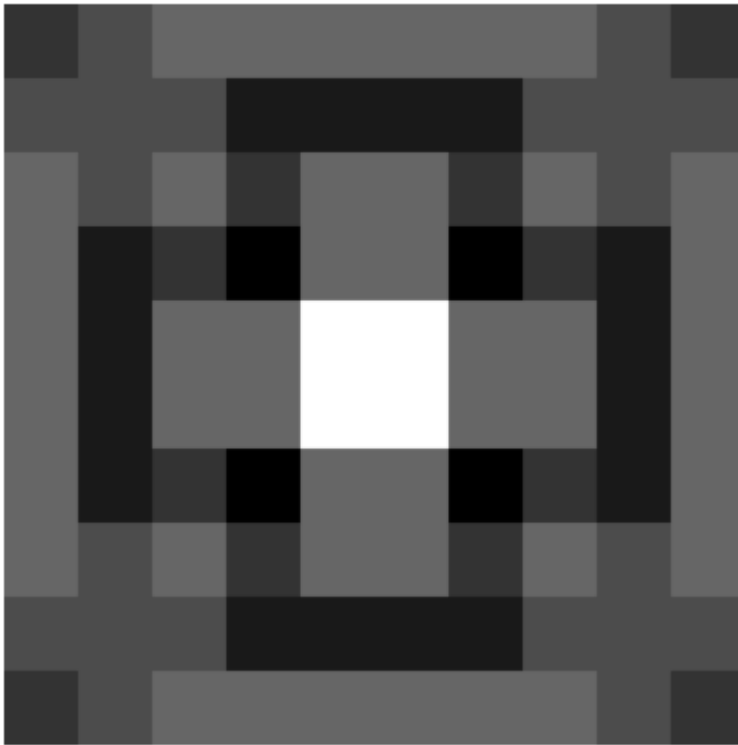
nearest



mirror



wrap

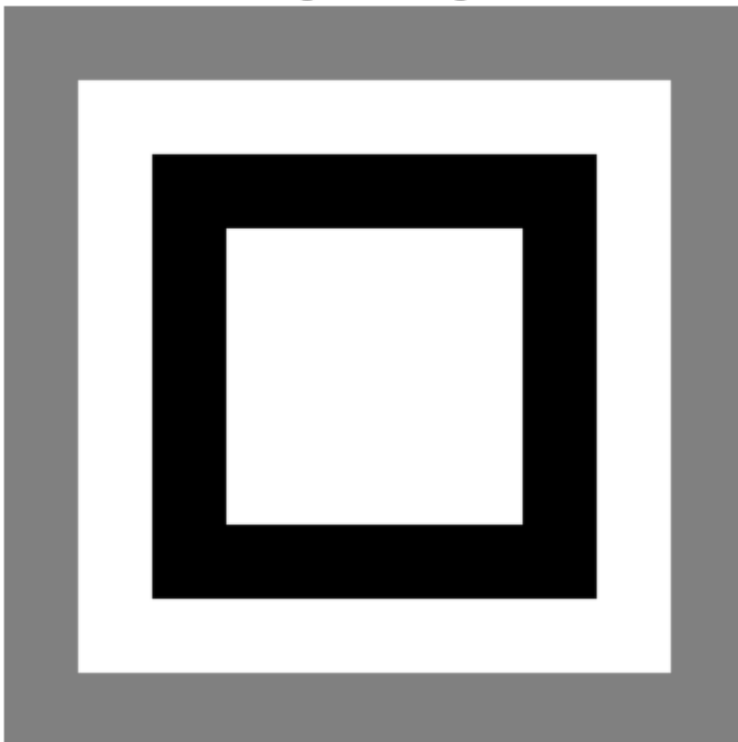


```
In [7]: print(f'\n\n5x5 averaging kernel')  
B = np.ones((5,5))/25  
ConvolveAlternatives(I,B)
```

5x5 averaging kernel

Original Image is 10 by 10

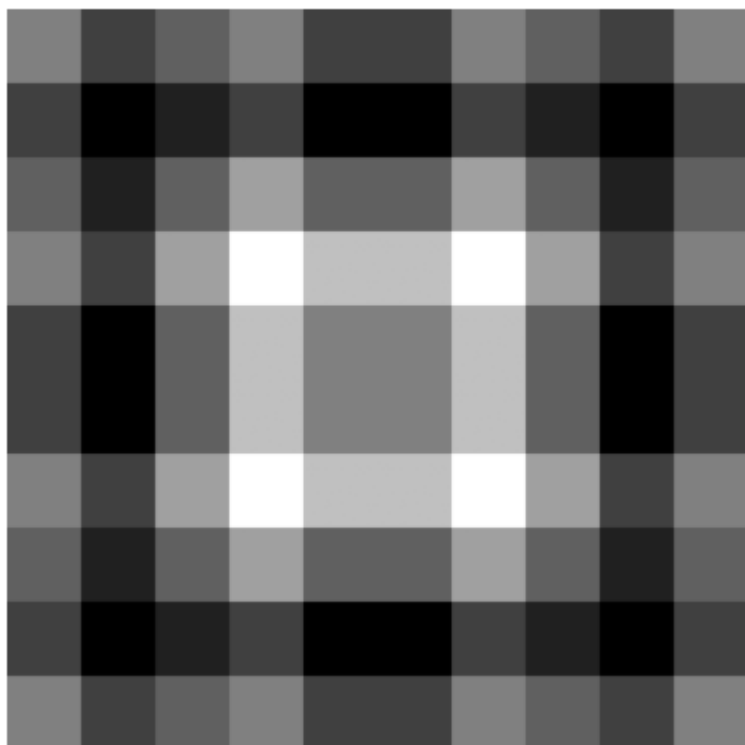
original image



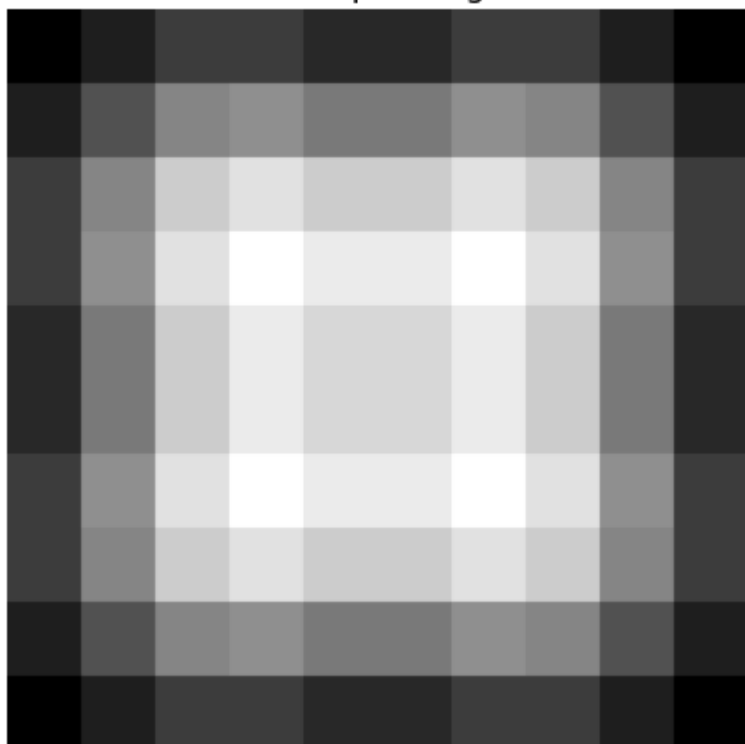
Kernel:

	0.04	0.04	0.04	0.04	0.04	
	0.04	0.04	0.04	0.04	0.04	
	0.04	0.04	0.04	0.04	0.04	
	0.04	0.04	0.04	0.04	0.04	
	0.04	0.04	0.04	0.04	0.04	

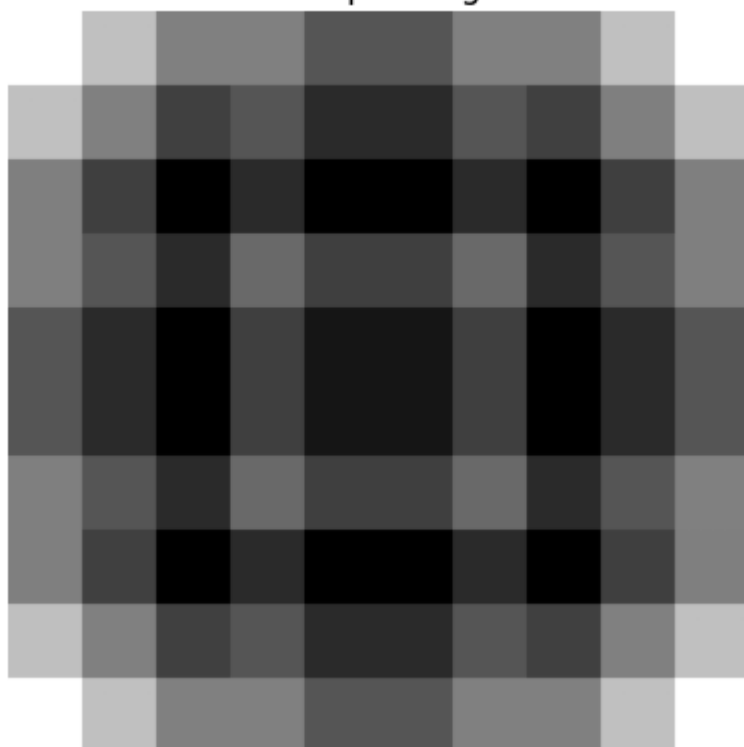
reflect



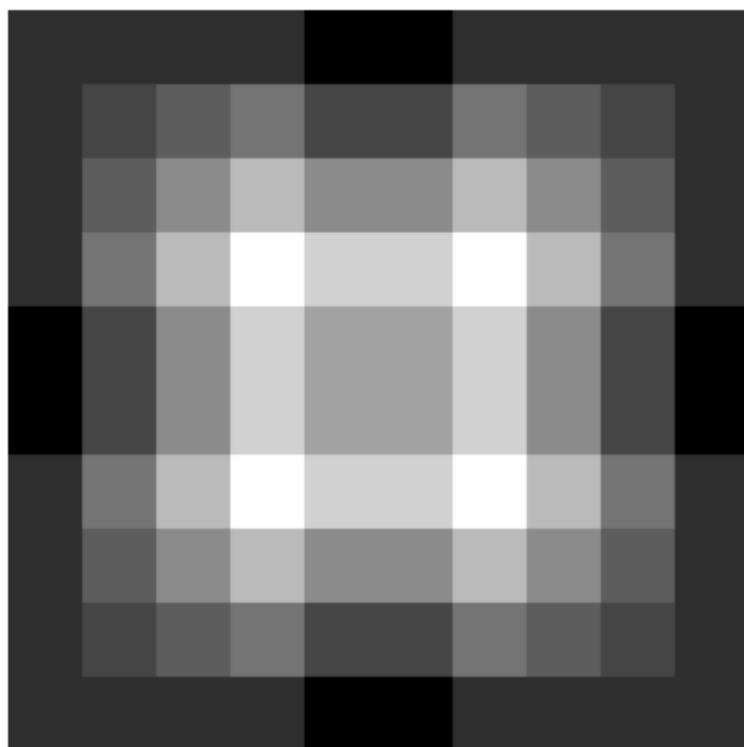
zero padding



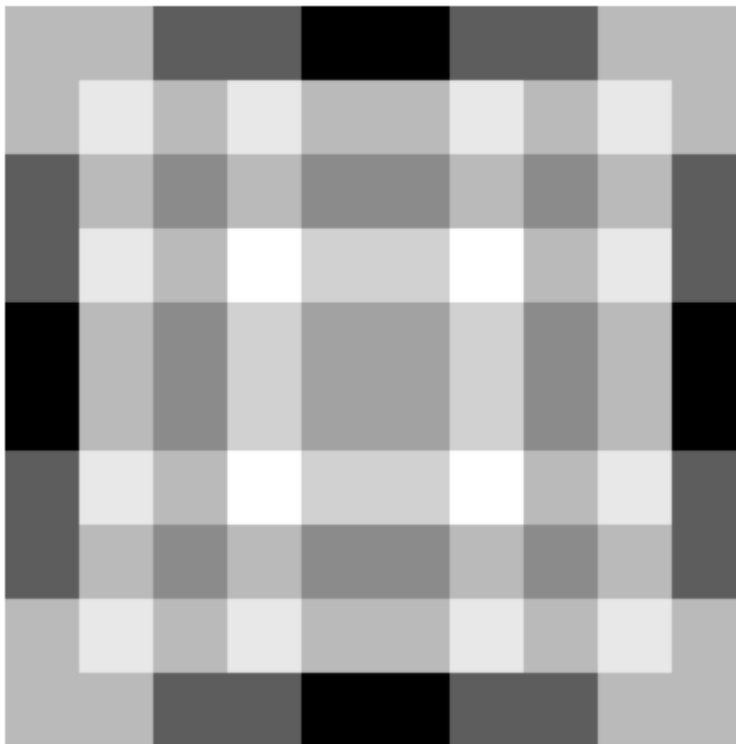
one padding



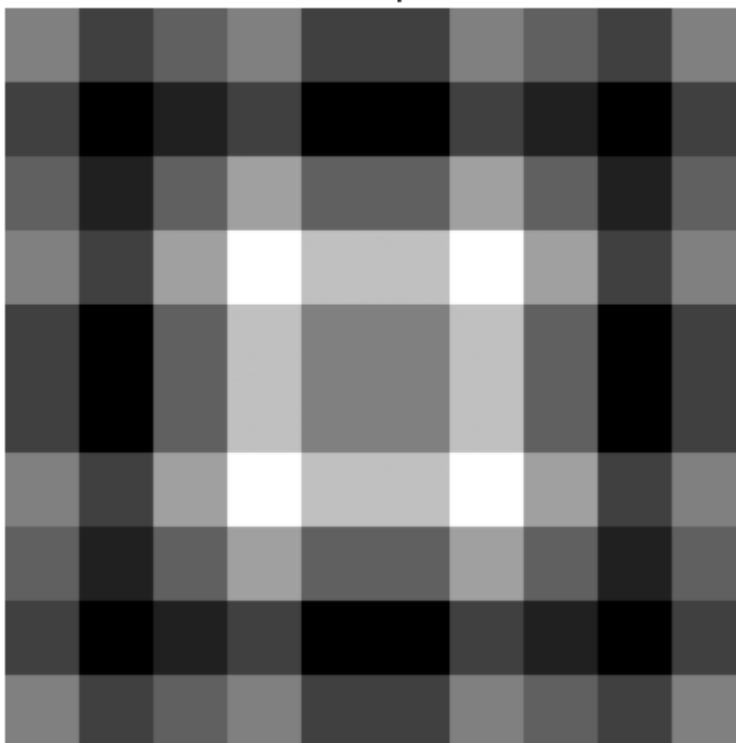
nearest



mirror



wrap



A weighted averaging kernel

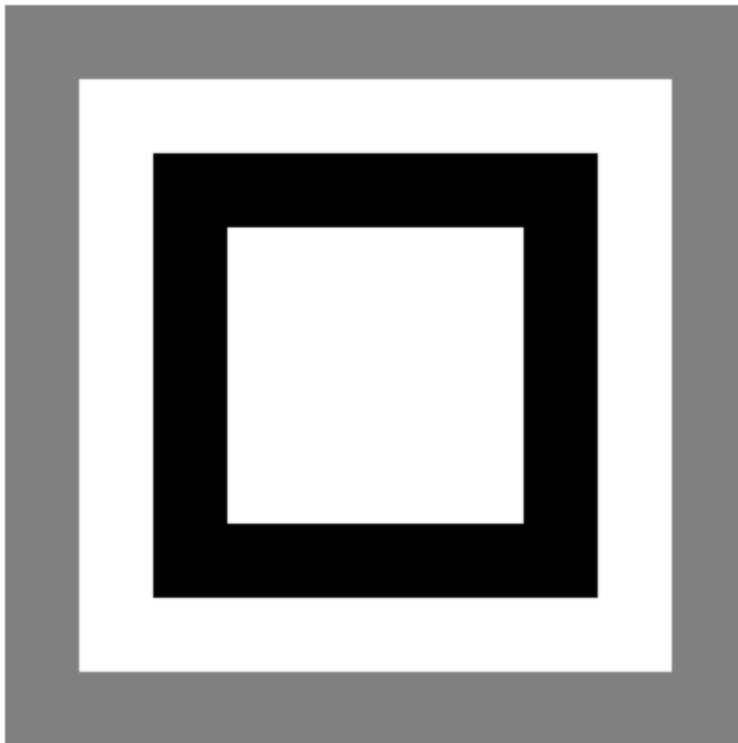
```
In [8]: # simple averaging Kernel
print(f'\n\n3x3 averaging kernel very heavy at the center')
B = np.ones((3,3))
B[1,:]=2
B[:,1]=2
B[1,1]=15
B = B / B.sum()
```

```
ConvolveAlternatives(I,B)
```

3x3 averaging kernel very heavy at the center

Original Image is 10 by 10

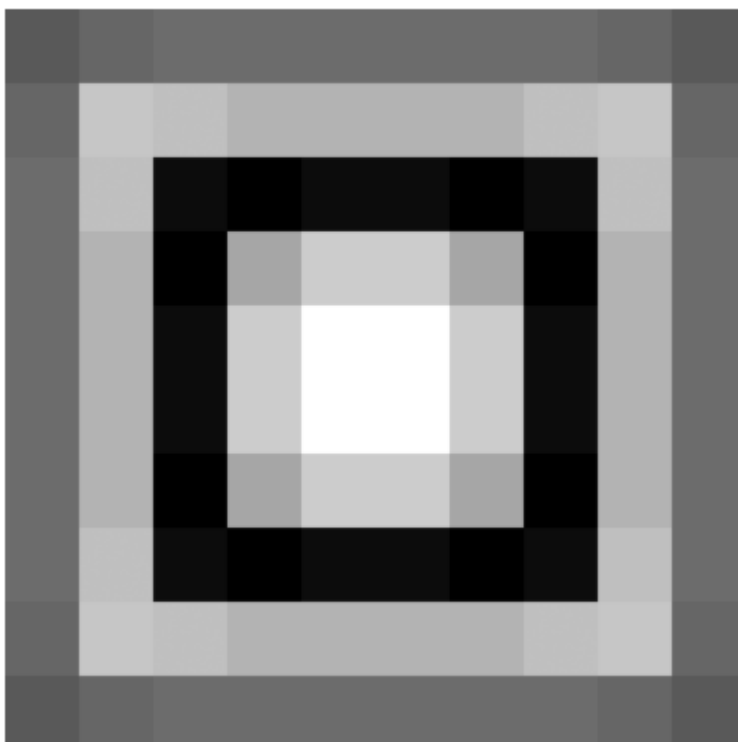
original image



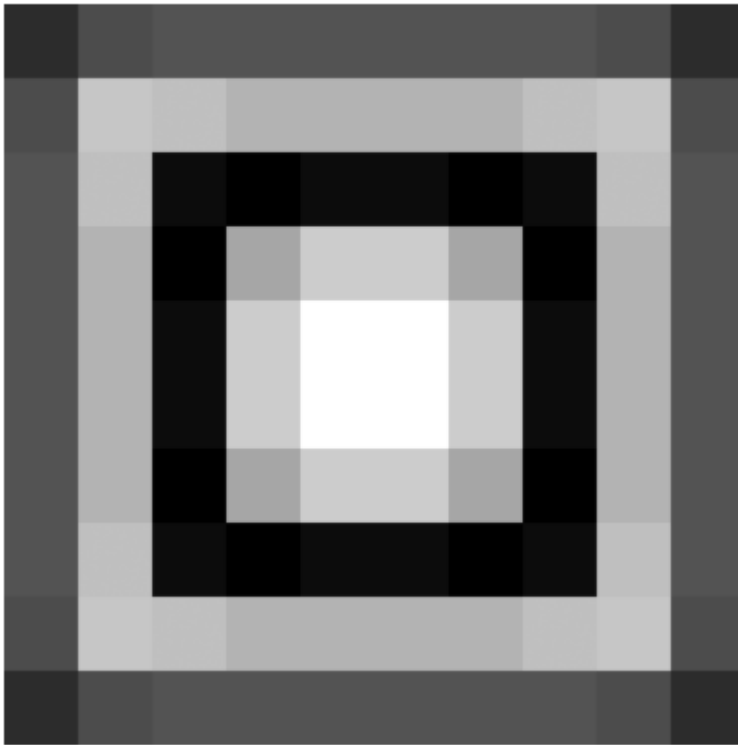
Kernel:

	0.04	0.07	0.04	
	0.07	0.56	0.07	
	0.04	0.07	0.04	

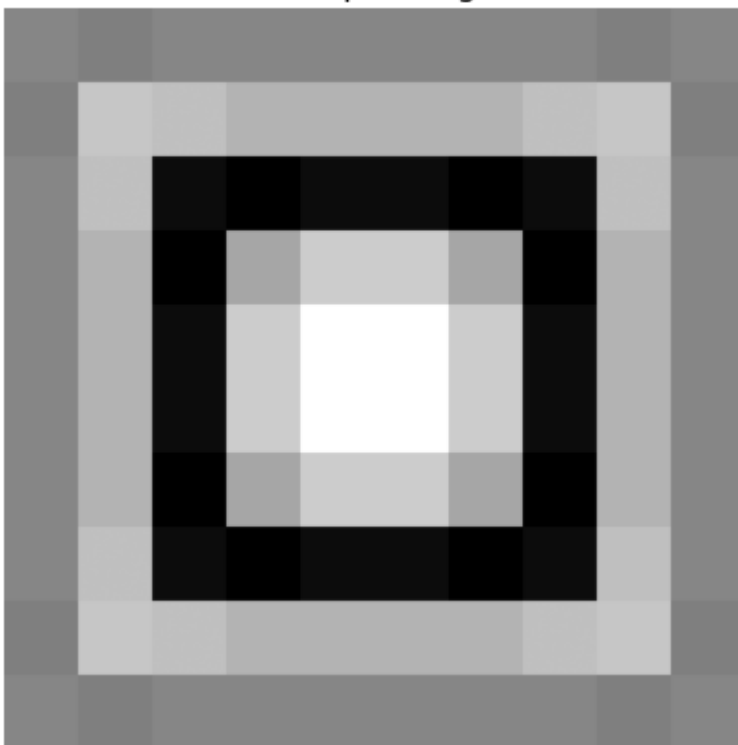
reflect



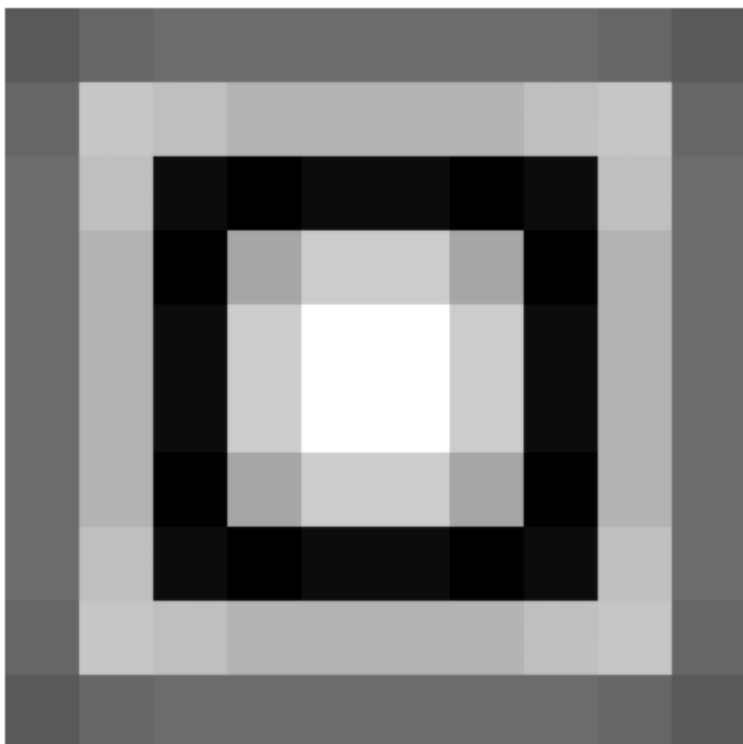
zero padding



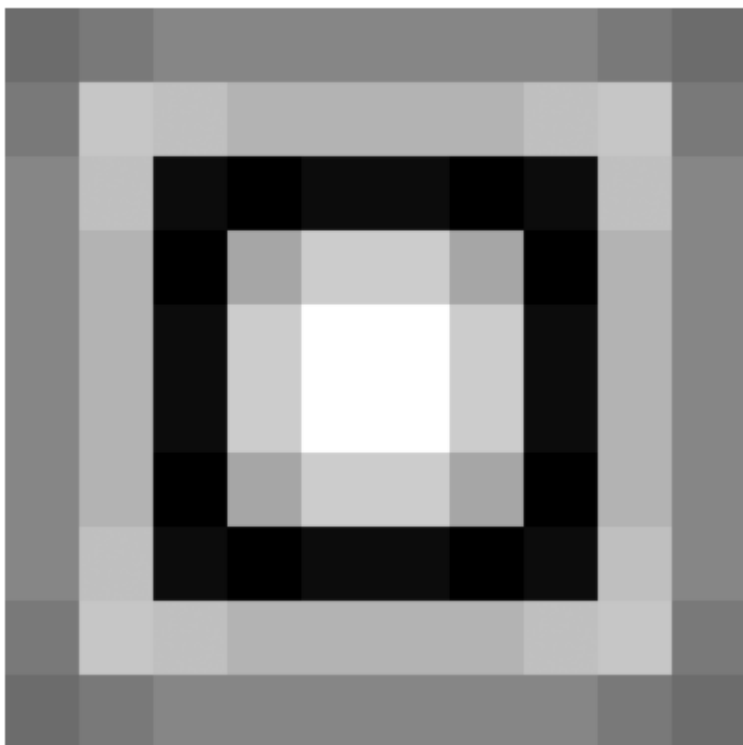
one padding

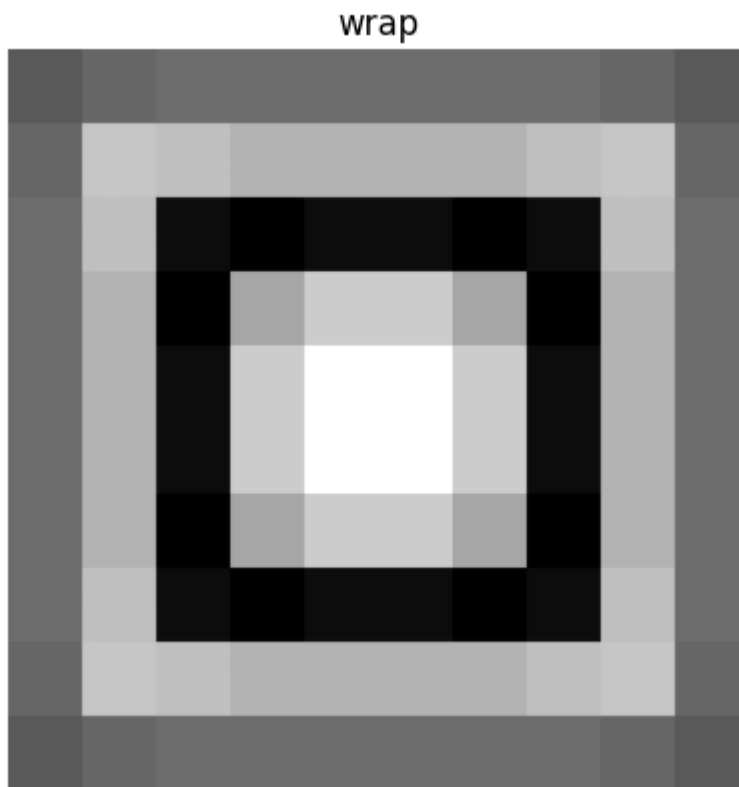


nearest



mirror





What does this kernel do?

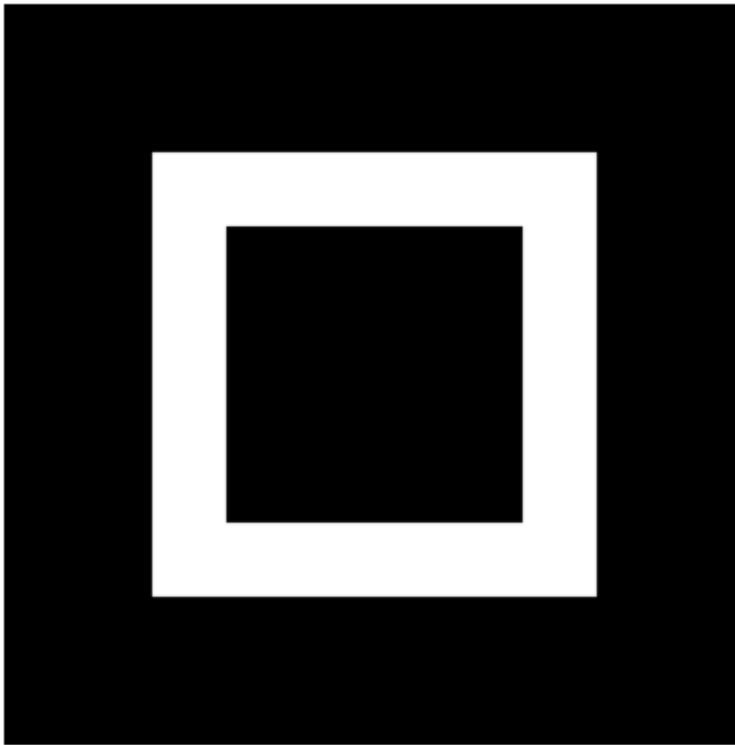
```
In [9]: # Generate a new test image
I = np.zeros((10,10))
I[(2,7),2:8] = 1
I[3:8,(2,7)] = 1
print(f'\n\n3x3 kernel with negative entries!!!')
B = np.array( [[-1, -1, -1], [2,2,2], [-1,-1,-1]] )
MatPrint(B, '\nKernel with negative entries')
IB = convolve(I,B)
IB = IB - IB.min()
IB = IB / IB.max()
MatImshow(I, 'Original Image')
dummy = MatImshow(convolve(I,B), 'Convolved Image')
```

3x3 kernel with negative entries!!!

Kernel with negative entries

	-1.00	-1.00	-1.00	
	2.00	2.00	2.00	
	-1.00	-1.00	-1.00	

Original Image



Convolved Image



why always 3x3

try 3×1 and 1×3

Don't they look like line detectors?

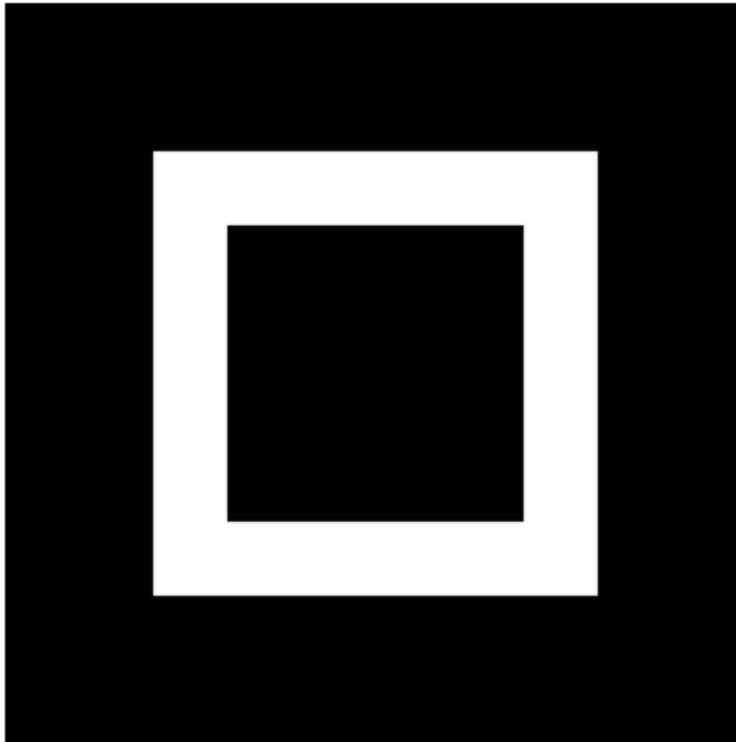
```
In [10]: # Generate a new test image
I = np.zeros((10,10))
I[(2,7),2:8] = 1
I[3:8,(2,7)] = 1
```

```

print(f'\n\n3x3 averaging kernel')
B = np.array( [[-1, 2, -1]] )
IB = convolve(I,B)
IB = IB - IB.min()
IB = IB / IB.max()
MatImshow(I, 'original image')
dummy = MatImshow(convolve(I,B), f'convolved with B:\n{B}')
dummy = MatImshow(convolve(I,B.T), f'convolved with B.T\n{B.T}')
```

3x3 averaging kernel

original image



convolved with B:
 $\begin{bmatrix} -1 & 2 & -1 \end{bmatrix}$



convolved with B.T
 $\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$



More kernels to play with

Base kernel **B0** is generated first, then its with respected to horizontal and vertical axes

along with its reflection are tested.

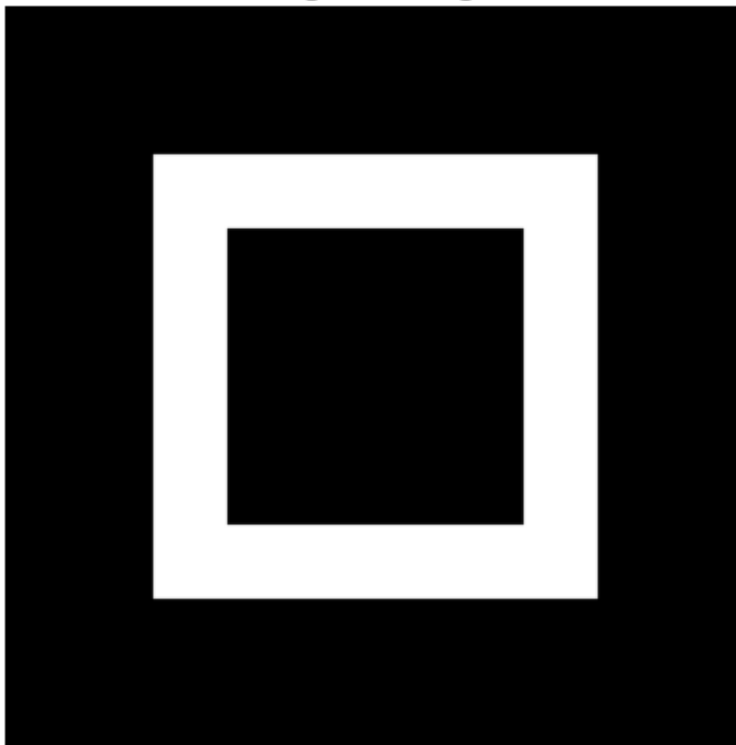
```
In [11]: # Generate a new test image
I = np.zeros((10,10))
I[(2,7),2:8] = 1
I[3:8,(2,7)] = 1
print(f'\n\n3x3 averaging kernel')
B0 = np.array( [[-1, -1, -1], [-1, 1, 1], [-1,1,-1]] )
B1 = B0[-1::-1,:] # horizontal flip of B0
B2 = B1[:, -1::-1] # vertical flip of B1
B3 = B2[-1::-1,:] # horizontal flip of B2

kernels = [B0, B1, B2, B3]

MatImshow(I, title='Original Image')
print('\n\nConvolved Images:')
for K in kernels:
    dummy = MatImshow(convolve(I,K), f'with kernel\n{K}')
```

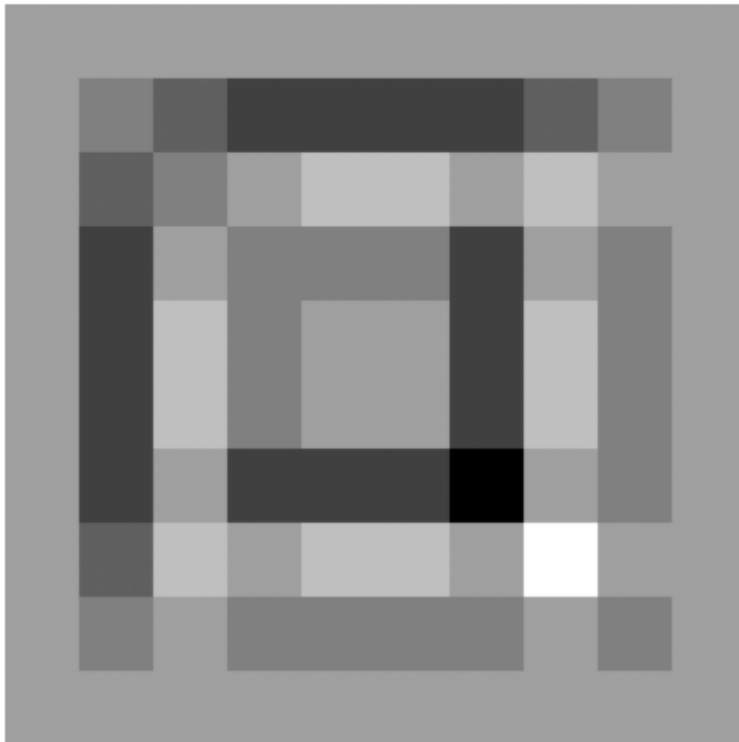
3x3 averaging kernel

Original Image

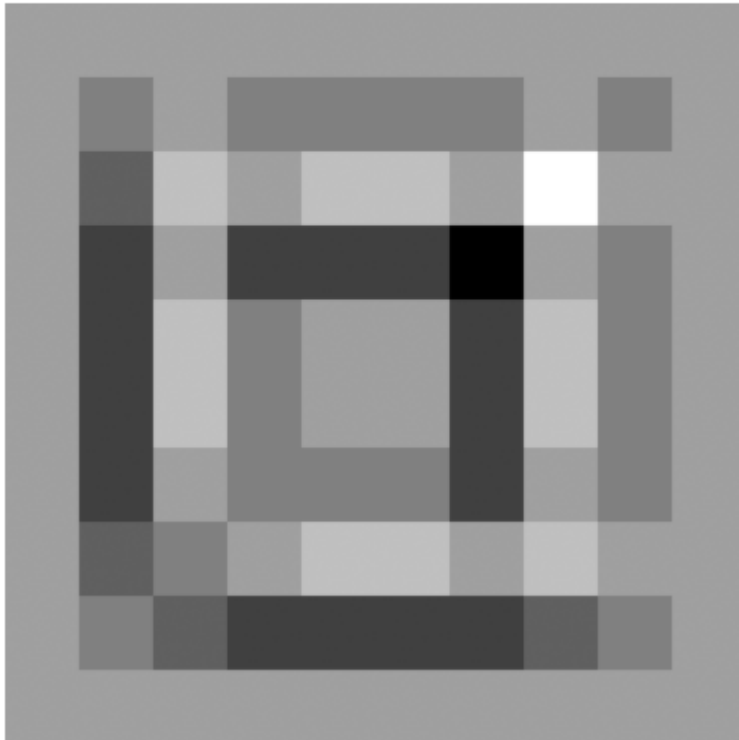


Convolved Images:

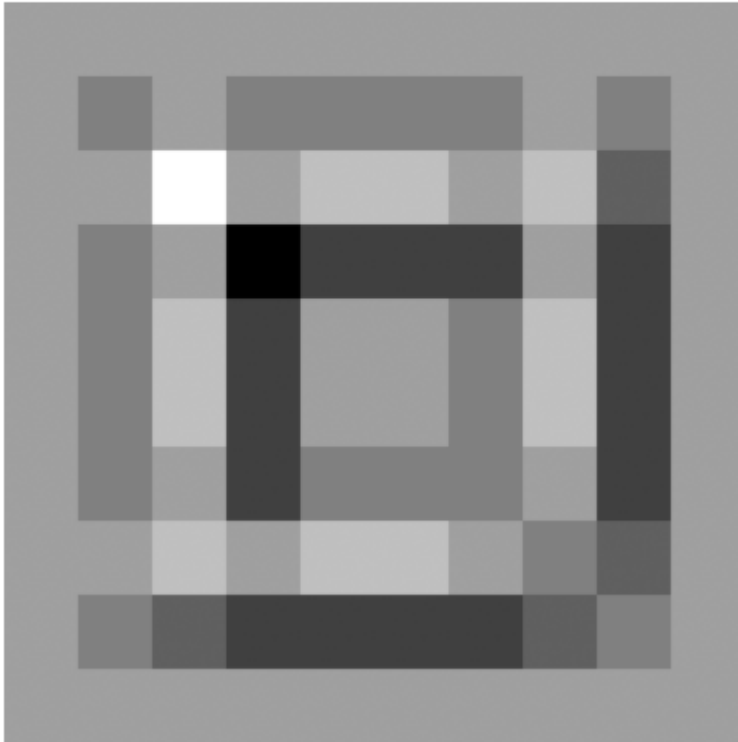
with kernel
[[-1 -1 -1]
[-1 1 1]
[-1 1 -1]]



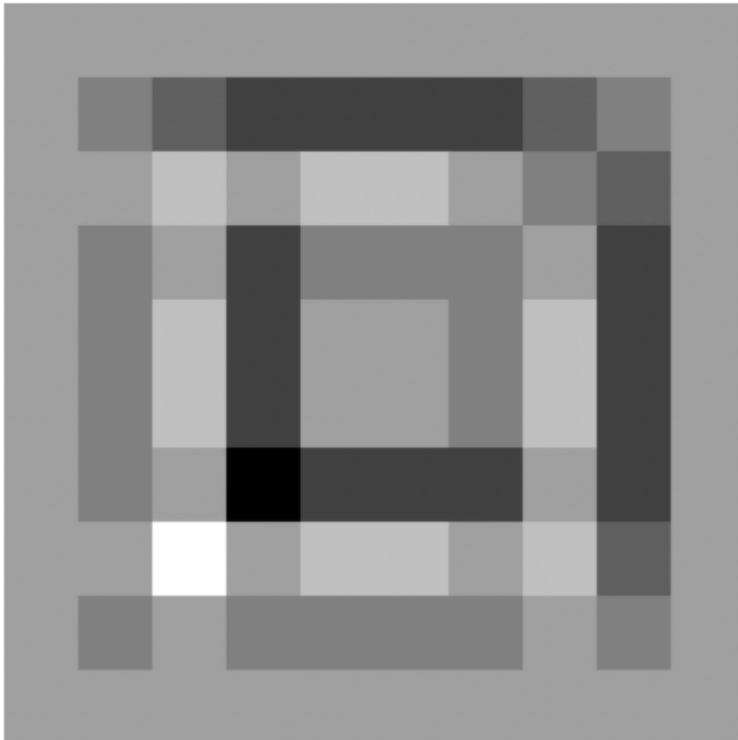
with kernel
[[-1 1 -1]
[-1 1 1]
[-1 -1 -1]]



with kernel
[[-1 1 -1]
[1 1 -1]
[-1 -1 -1]]



with kernel
[[-1 -1 -1]
[1 1 -1]
[-1 1 -1]]

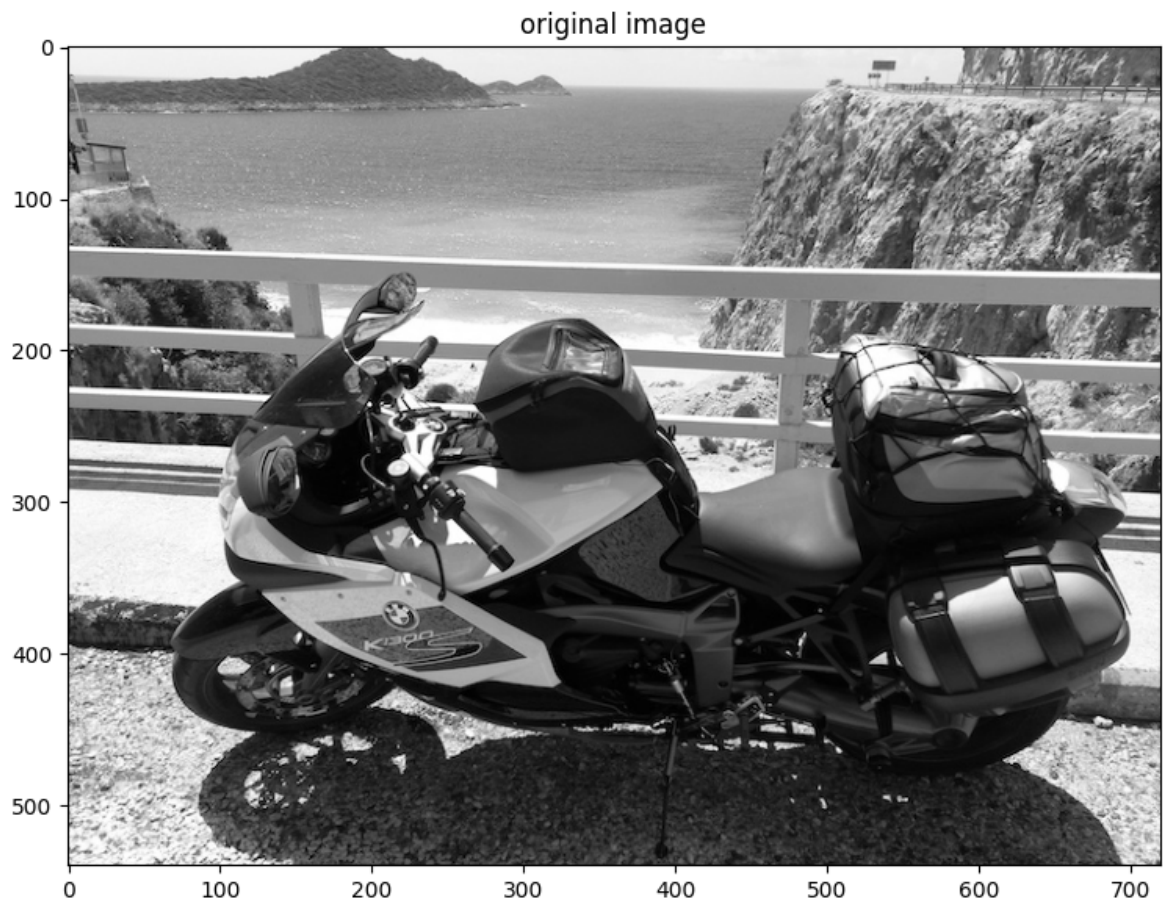


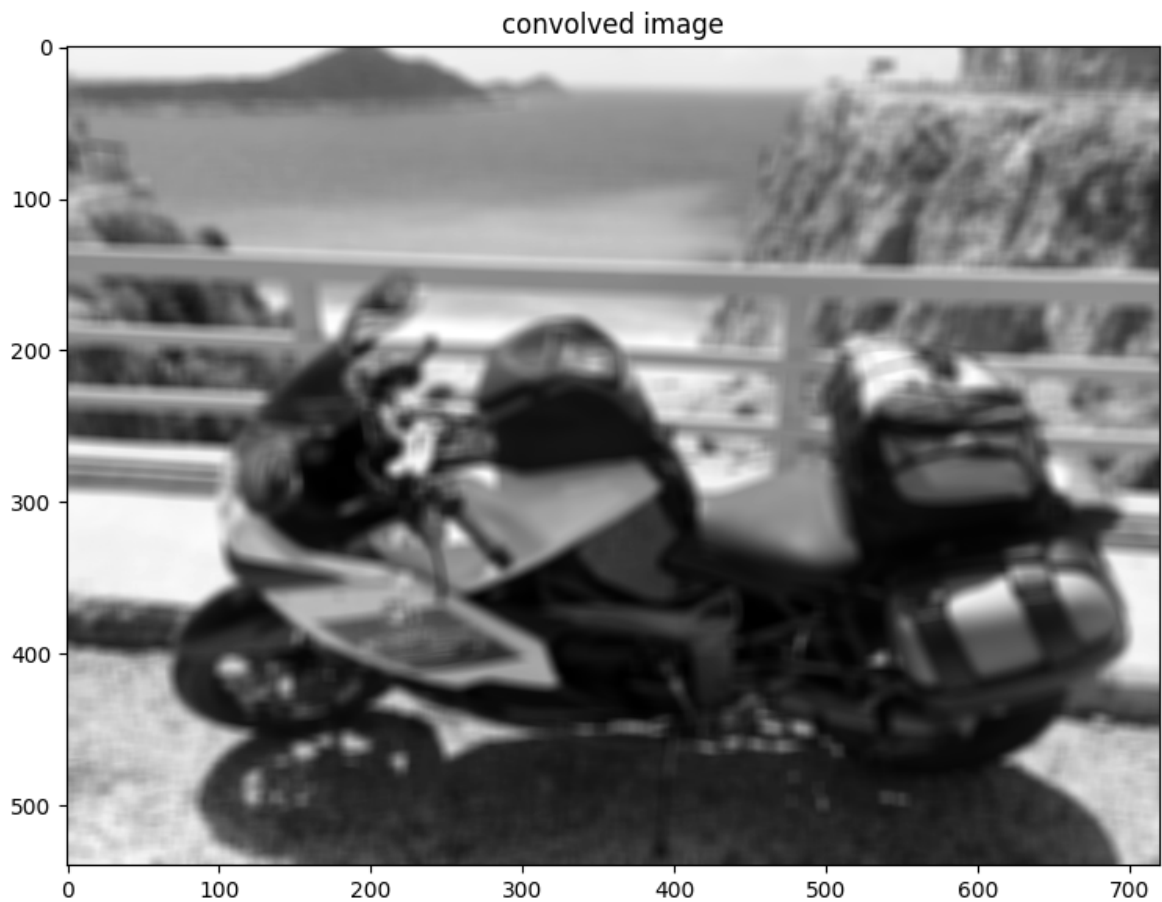
let's play on some more realistic images

Start with gray scale

```
In [12]: fSize = (9,9)
c0 = io.imread('con2.jpg', as_gray=True)
plt.figure(figsize=fSize)
plt.imshow(c0, cmap='gray')
plt.title('original image')
filter_size = 9
b = np.ones((filter_size,filter_size)) / filter_size**2

c0c = convolve(c0, b)
plt.figure(figsize=fSize)
plt.imshow(c0c, cmap='gray')
plt.title('convolved image')
plt.show()
```





How about color

```
In [13]: fSize = (9,9)
c1 = io.imread('con3.jpg')
plt.figure(figsize=fSize)
plt.imshow(c1)
plt.title('original image')
filter_size = 7
b = np.ones((filter_size,filter_size)) / filter_size**2

clc = c1.copy()
# blur each layer, note that you can blur each layer with different kernels
for i in range(c1.shape[2]):
    clc[:, :, i] = convolve(c1[:, :, i], b)
plt.figure(figsize=fSize)
plt.imshow(clc)
plt.title('convolved image')
plt.show()
```

original image



convolved image



collection of convolved images

What can we do using a collection of kernels?

```

In [14]: # Generate a new test image
I = np.zeros((10,10))
I[(2,7),2:8] = 1
I[3:8,(2,7)] = 1
print(f'\n\n3x3 averaging kernel')
B0 = np.array( [[-1, -1, -1], [-1, 1, 1], [-1,1,-1]] )
B1 = B0[-1::-1,:] # horizontal flip of B0
B2 = B1[:, -1::-1] # vertical flip of B1
B3 = B2[-1::-1,:] # horizontal flip of B2
Bv = np.array( [[-1, 2, -1]] )
Bh = Bv.T

B = [B0, B1, B2, B3, Bv, Bh]
print('Print kernels')
for i, K in enumerate(B):
    MatPrint(K, f'Kernel {i+1}')
    print('\n ')

print('Show kernels as binary images')
for K in B:
    MatImshow(K, f'Kernel:\n{K}')
    print('\n ')

res = np.empty((I.shape[0], I.shape[1], len(B)))
csum = np.zeros_like(I)
for ind, K in enumerate(B): #go over kernels
    r = convolve(I, K) # convolve image with the current kernel
    csum += r # add current image to running sum... just to see what happ
    MatImshow(r, f'Image convolved with:\n{K}') # display the current fil
    rReLUd = np.where(r<0, np.zeros_like(r), r)
    MatImshow(rReLUd, f'Feature Map after ReLu')
    print('\n ')
    res[:, :, ind] = r
    #res = np.append(res, r, axis=2)
MatImshow(csum)
MatImshow(res.max(axis=2))

```

3x3 averaging kernel

Print kernels

Kernel 1

	-1.00	-1.00	-1.00	
	-1.00	1.00	1.00	
	-1.00	1.00	-1.00	

Kernel 2

	-1.00	1.00	-1.00	
	-1.00	1.00	1.00	
	-1.00	-1.00	-1.00	

Kernel 3

	-1.00	1.00	-1.00	
	1.00	1.00	-1.00	
	-1.00	-1.00	-1.00	

Kernel 4

	-1.00	-1.00	-1.00	
	1.00	1.00	-1.00	
	-1.00	1.00	-1.00	

Kernel 5

	-1.00	2.00	-1.00	
--	-------	------	-------	--

Kernel 6

	-1.00	
	2.00	
	-1.00	

Show kernels as binary images

Kernel:
[[-1 -1 -1]
[-1 1 1]
[-1 1 -1]]



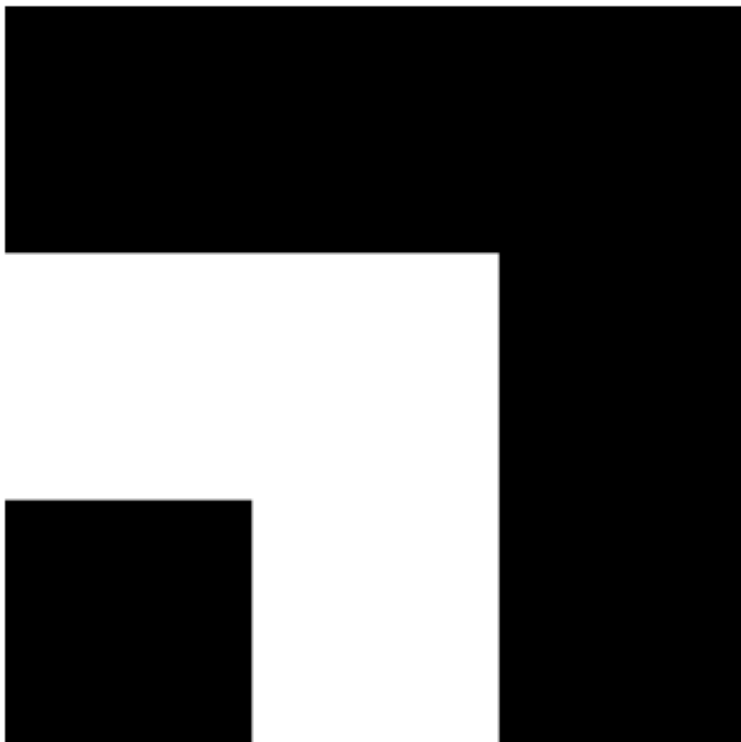
Kernel:
[[-1 1 -1]
[-1 1 1]
[-1 -1 -1]]



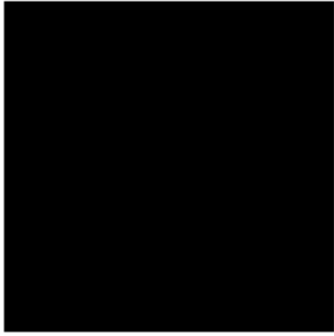
Kernel:
[[-1 1 -1]
[1 1 -1]
[-1 -1 -1]]



Kernel:
[[-1 -1 -1]
[1 1 -1]
[-1 1 -1]]



Kernel:
[[-1 2 -1]]

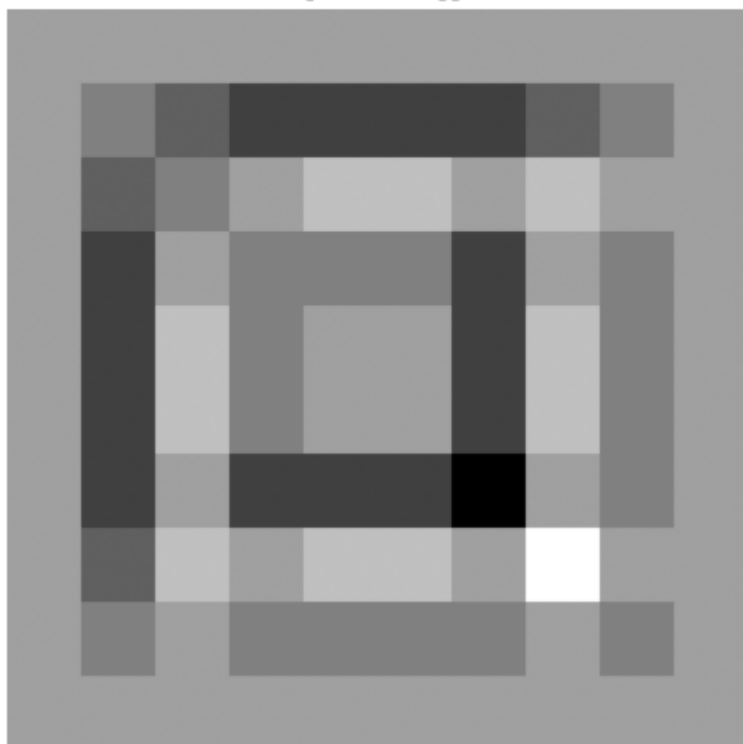


Kernel:
[[-1]
[2]
[-1]]



Image convolved with:

```
[[ -1 -1 -1]  
[ -1  1  1]  
[ -1  1 -1]]
```



Feature Map after ReLu

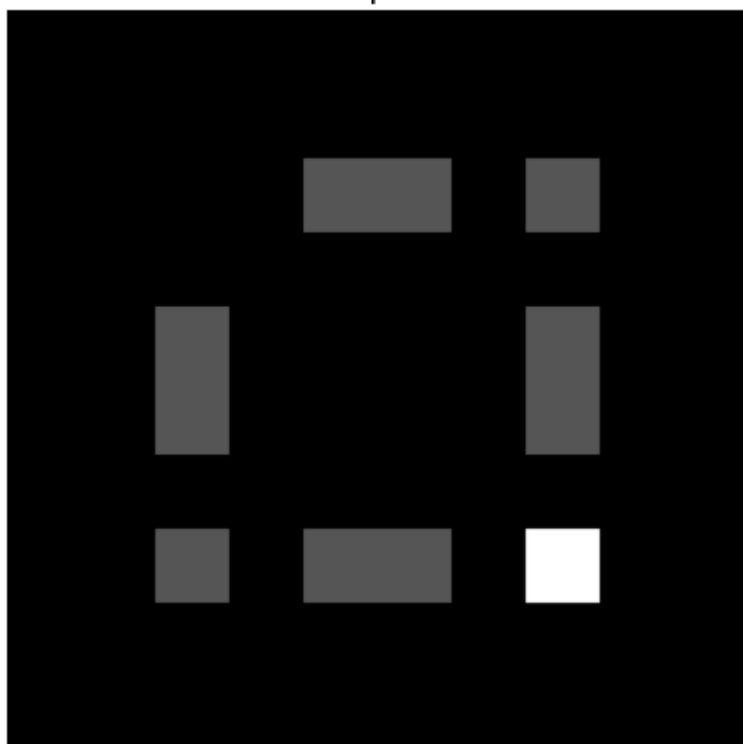
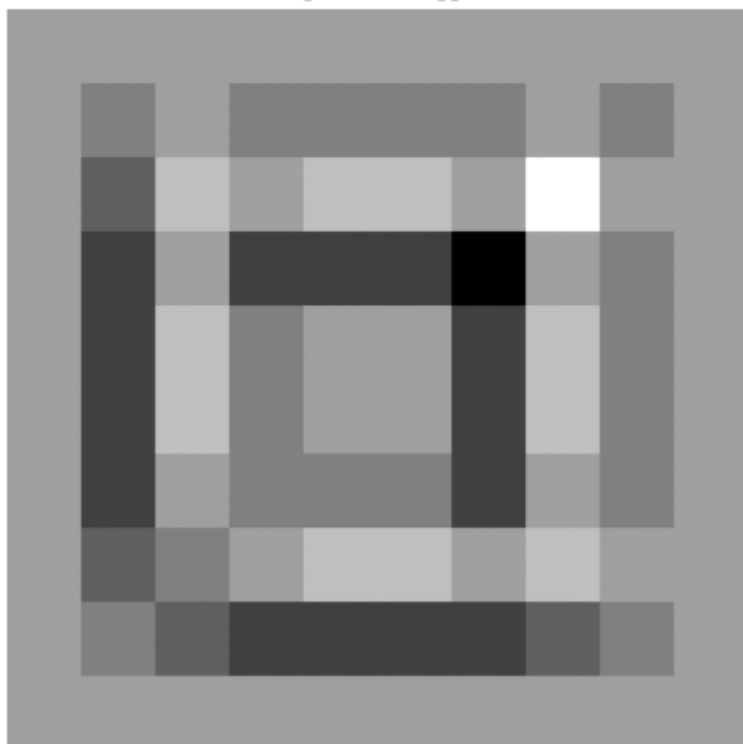


Image convolved with:

$\begin{bmatrix} -1 & 1 & -1 \end{bmatrix}$

$\begin{bmatrix} -1 & 1 & 1 \end{bmatrix}$

$\begin{bmatrix} -1 & -1 & -1 \end{bmatrix}$



Feature Map after ReLu

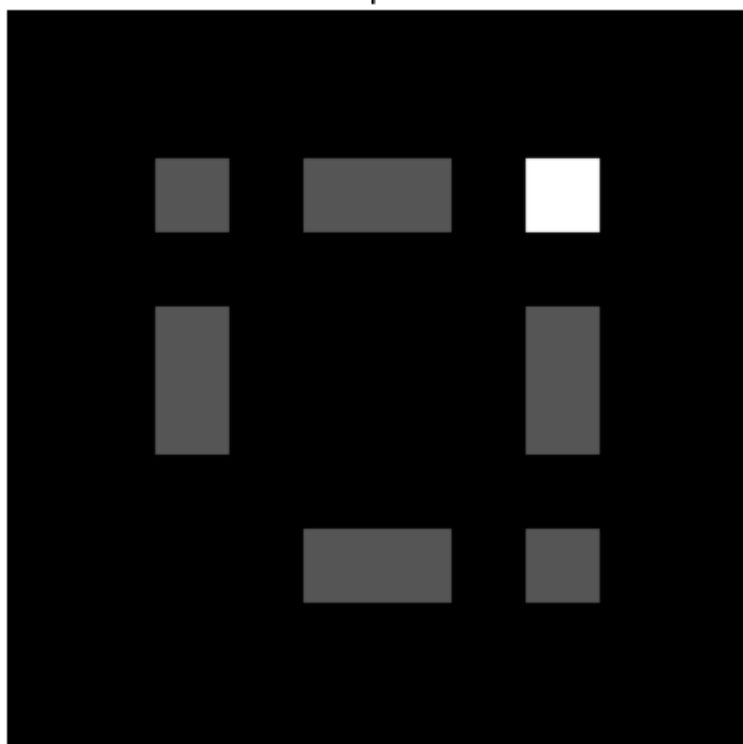
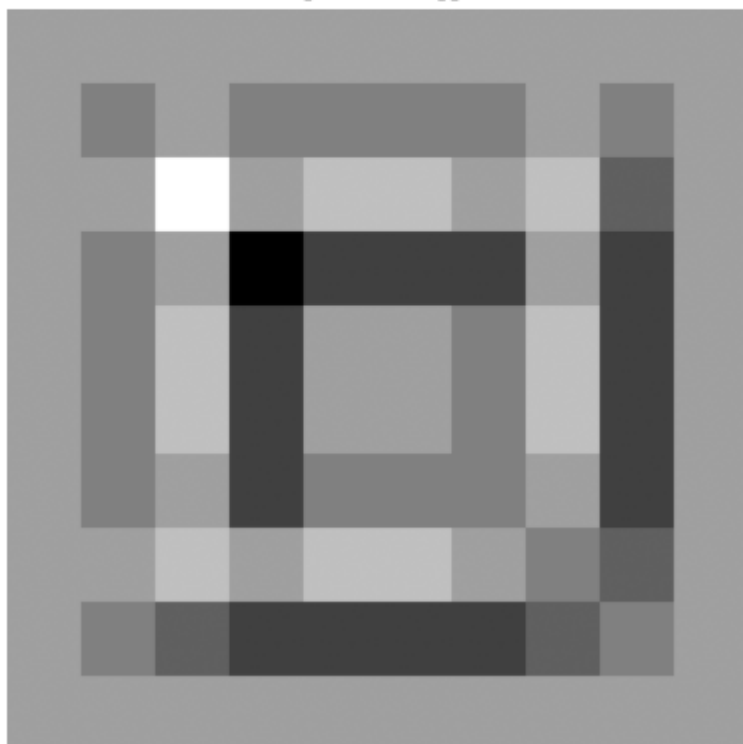


Image convolved with:

$\begin{bmatrix} -1 & 1 & -1 \end{bmatrix}$

$\begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$

$\begin{bmatrix} -1 & -1 & -1 \end{bmatrix}$



Feature Map after ReLu

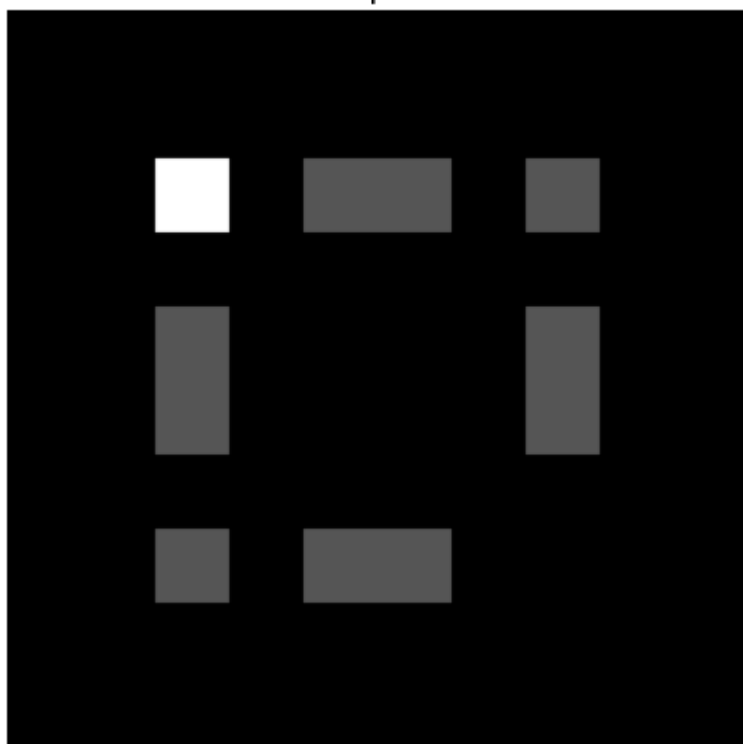
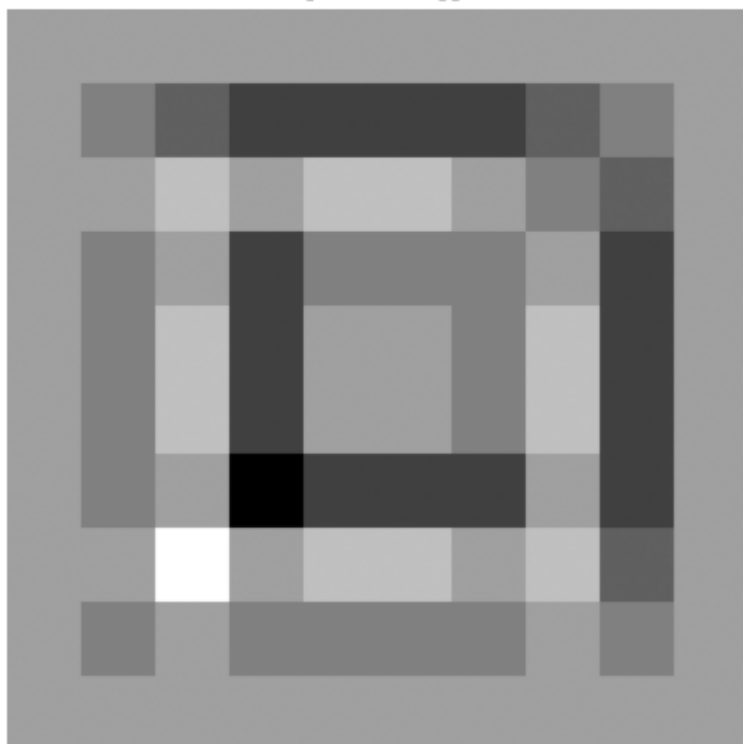


Image convolved with:

```
[[ -1 -1 -1]  
 [  1  1 -1]  
[-1  1 -1]]
```



Feature Map after ReLu

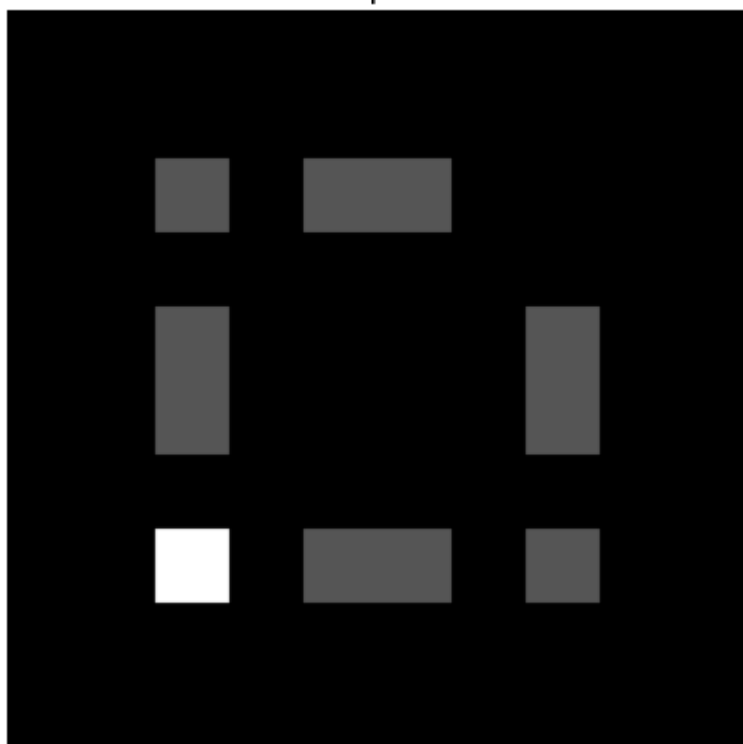


Image convolved with:
[[-1 2 -1]]



Feature Map after ReLu

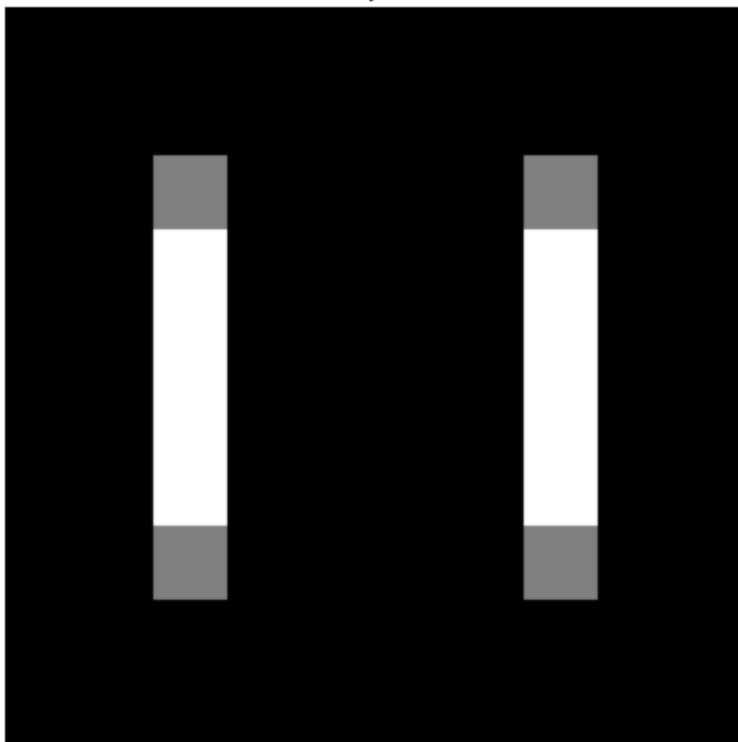
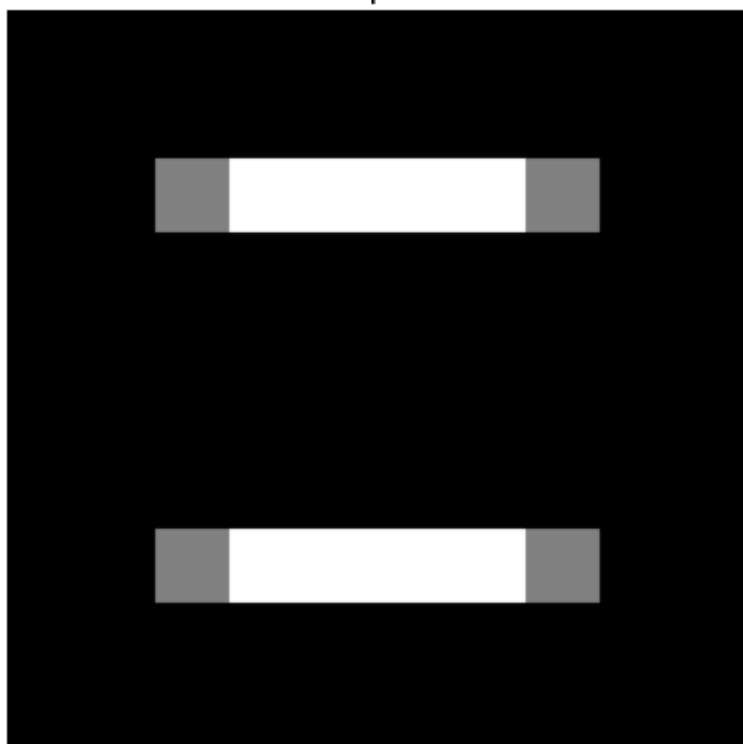


Image convolved with:

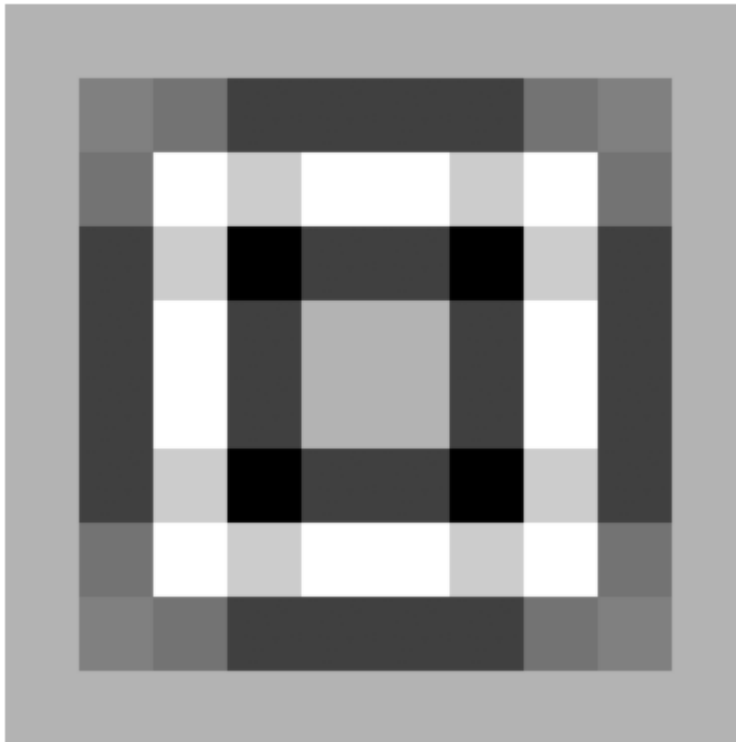
$\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$



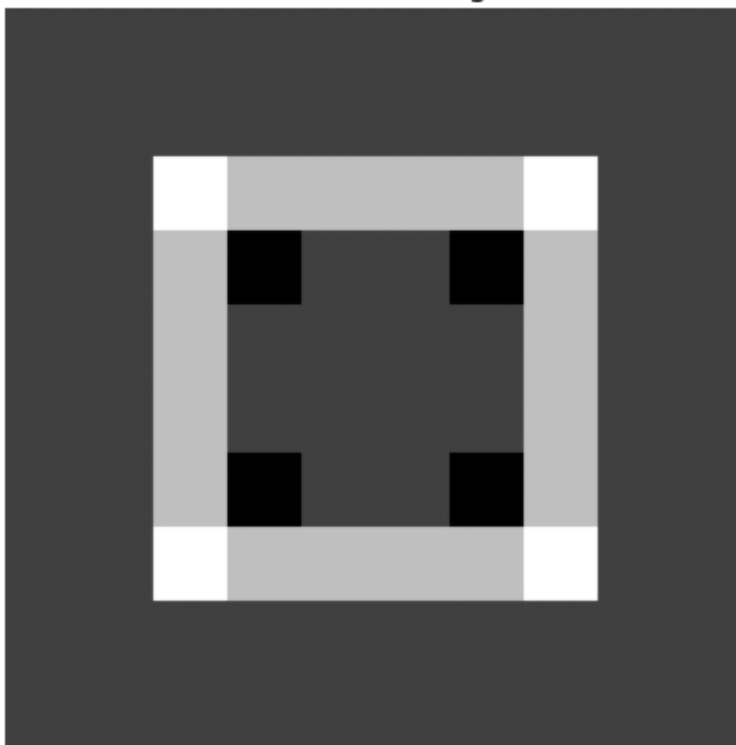
Feature Map after ReLu



Matrix as Image



Matrix as Image



```
Out[14]: array([[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]])
```

In []: