Spatial Correlation and Convolution

- There are two closely related concepts that must be understood when performing linear spatial filtering, One is *correlation* and the other is *convolution.*
- Correlation is the process of moving a filter mask over the image and computing the sum of products at each location.

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)$$

- The mechanics of convolution are the same, except that the filter is first rotated by 180°.

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t)$$

```
>> x = [0 0 0 1 0 0 0 0];
>> w = [1 2 3 2 8];
>> flip(w)
ans =
   8    2    3    2    1
```

Use Convolution to implement Correlation

```
>> conv(x, flip(w))

ans =
 Columns 1 through 11
   0    0    0    8    2    3    2    1    0    0    0
 Column 12
   0
```

```
>> conv(x, flip(w), 'same')
ans =
   0    8    2    3    2    1    0    0
```

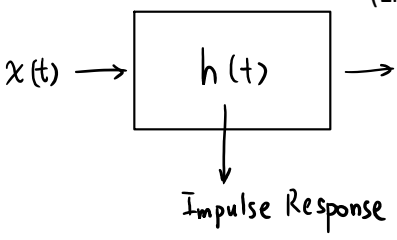
Cropped correlation result

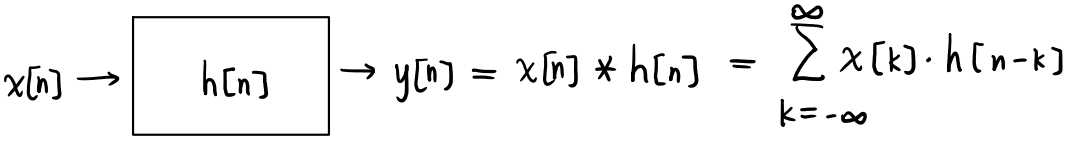
0 8 2 3 2 1 0 0

Full correlation result

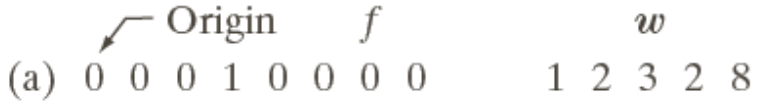
(g) 0 0 0 8 2 3 2 1 0 0 0 0

LTI System (Linear Time-Invariant)





If $x[n] = \delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$, then $y[n] = \delta[n] * h[n] = \sum_{k=-\infty}^{\infty} \delta[k] \cdot h[n-k] = h[n]$



Cropped convolution result

0 1 2 3 2 8 0 0

w

```
>> conv(x, w, 'same')

ans =
   0   1   2   3   2   8   0   0
```

Extended to 2D operations:

```
>> f = zeros(5,5);
>> f(3,3) = 1;
>> f
f =
   0   0   0   0   0
   0   0   0   0   0
   0   0   1   0   0
   0   0   0   0   0
   0   0   0   0   0
```

```
>> w = reshape(1:9, 3, 3)'

w =
   1   2   3
   4   5   6
   7   8   9
```

```
>> conv2(f, w, 'same')
ans =
   0   0   0   0   0
   0   1   2   3   0
   0   4   5   6   0
   0   7   8   9   0
```

Cropped convolution result

0 0 0 0 0
0 1 2 3 0
0 4 5 6 0
0 7 8 9 0

0 1 2 3 0
0 4 5 6 0
0 7 8 9 0
0 0 0 0 0

0 4 5 6 0
0 7 8 9 0
0 0 0 0 0

w =

1	2	3
4	5	6
7	8	9

>> w1 = flip(w, 1)

w1 =

7	8	9
4	5	6
1	2	3

>> w2 = flip(w1, 2)

w2 =

9	8	7
6	5	4
3	2	1

>> conv2(f, w2, 'same')

ans =

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

flip(A,1) reverses the elements in each column, and flip(A,2) reverses the elements in each row.

Boundary Padding Options

>> A = [1 2; 3 4]

A =

1	2
3	4

>> padarray(A, [2 2])

Zero Padding

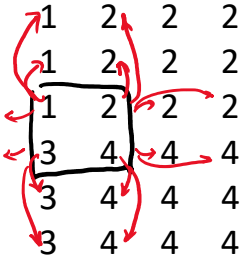
ans =

0	0	0	0	0	0
0	0	0	0	0	0
0	0	1	2	0	0
0	0	3	4	0	0
0	0	0	0	0	0
0	0	0	0	0	0

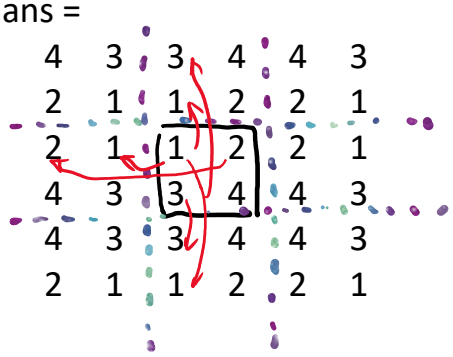
>> padarray(A, [2 2], 'replicate')

ans =

1	1	1	2	2	2
1	1	1	2	2	2
1	1	1	2	2	2
3	3	3	4	4	4
3	3	3	4	4	4
3	3	3	4	4	4



```
>> padarray(A, [2 2], 'symmetric')
ans =
   4   3   3   4   4   3
   2   1   1   2   2   1
   2   1   1   2   2   1
   4   3   3   4   4   3
   4   3   3   4   4   3
   2   1   1   2   2   1
```



```
>> padarray(A, [2 2], 'circular')
ans =
   1   2   1   2   1   2
   3   4   3   4   3   4
   1   2   1   2   1   2
   3   4   3   4   3   4
   1   2   1   2   1   2
   3   4   3   4   3   4
```

- Smoothing Filter

```
>> I = imread('Fig0335(a)
(ckt_board_saltpep_prob_pt05).tif');
>> imshow(I)
>> h = ones(3,3)/9;
>> h
h =
   0.1111   0.1111   0.1111
   0.1111   0.1111   0.1111
   0.1111   0.1111   0.1111
```

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

```
>> J = imfilter(I, h, 'symmetric');
>> figure; imshowpair(I,J,'montage')
```

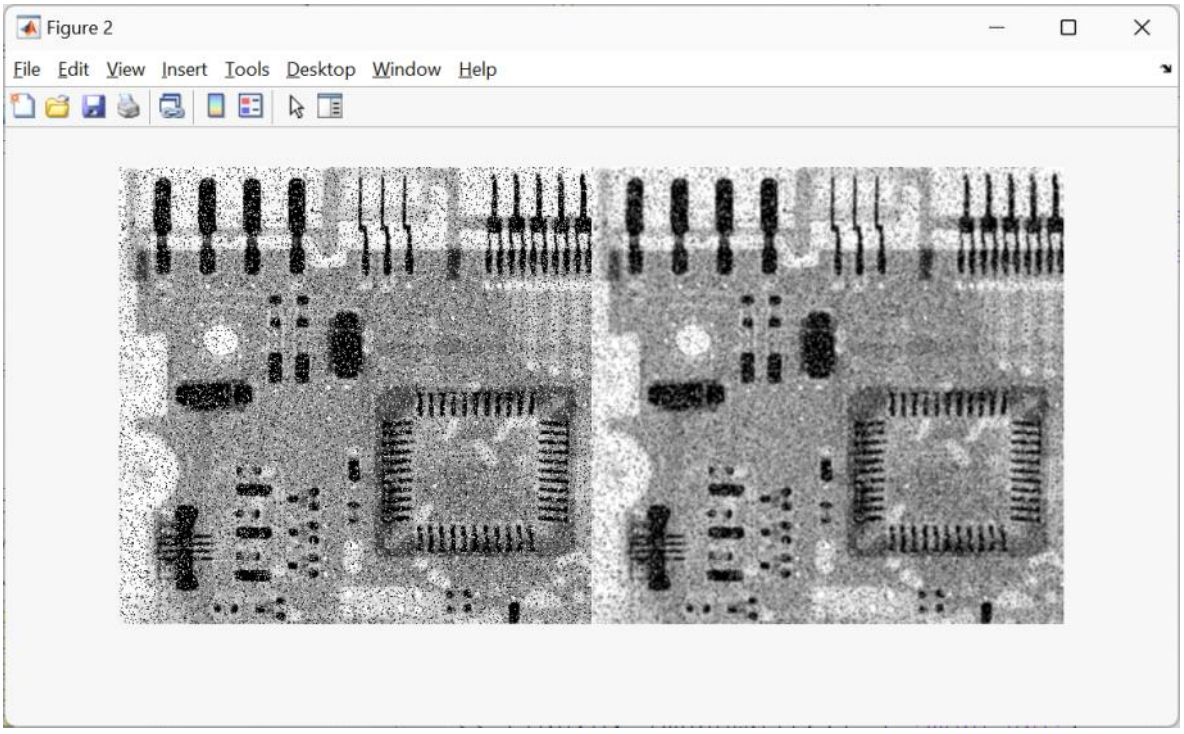
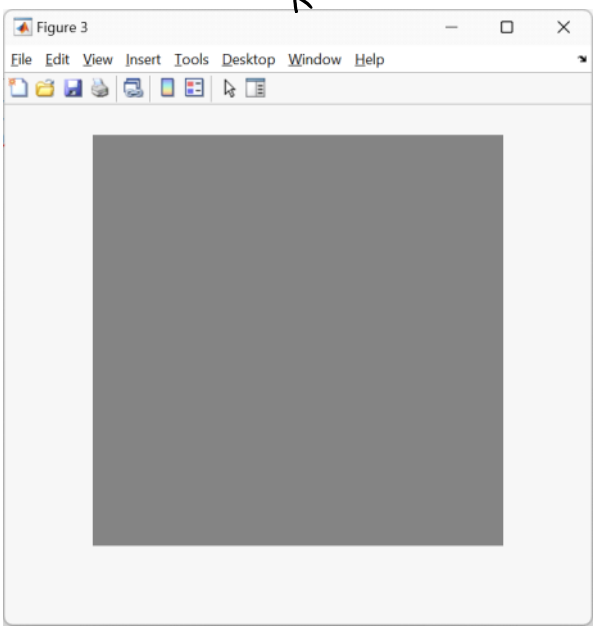


Figure 2
File Edit View Insert Tools Desktop Window Help

```
>> K = uint8(128*ones(512, 512));
>> figure; imshow(K)
>> S = imfiter(K, h, 'symmetric');
```

K



S