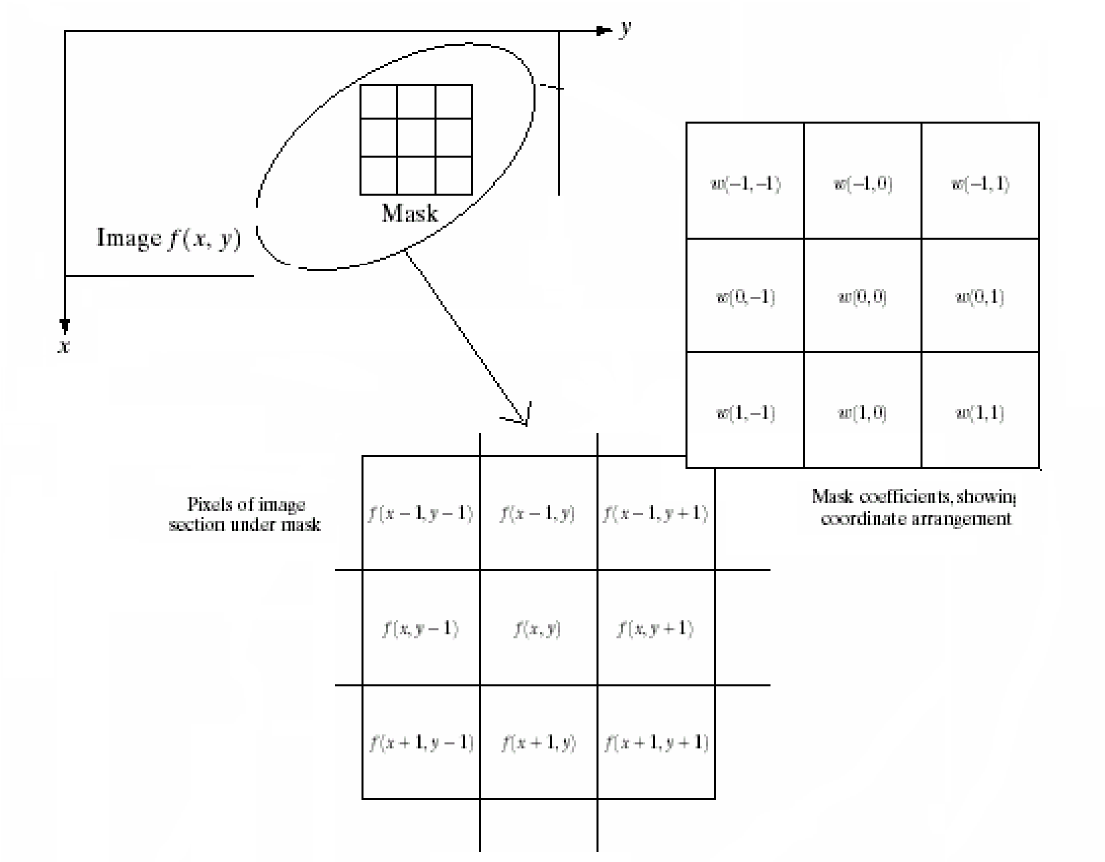
Convolution and Spatial Filtering

- **Linear Spatial Filtering**
- The most commonly used type of neighborhood operator is a linear filter, in which an output pixel's value is determined as a weighted sum of input pixel values.
- Filtering in the spatial domain refers to image operators that transform the gray value at a pixel (x,y) in terms of the pixel values in a neighbouring square window centred at (x,y) using a given **matrix** of the same size.
- The matrix is called a **filter**, **mask**, **kernel** or a **window** (h in the below equation). The operation is mainly the inner product (also known as the **convolution**) of the neighbouring sub image with the filter.
- $$\sum_{k,l} f(k,l)h(i-k,j-l),$$

$$g = f * h,$$

- Where g: resulting image, f : source image, h : kernel
- The filtering process works by replacing each pixel value with the result of convolution at the pixel.

- Filtering is often used to remove noise in images that could occur as a result of less than perfect imaging devices, signal interference, or even as a result of image processing such as HE transforms.



Convolution

- Example of neighborhood filtering (convolution).

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

$f(x,y)$

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

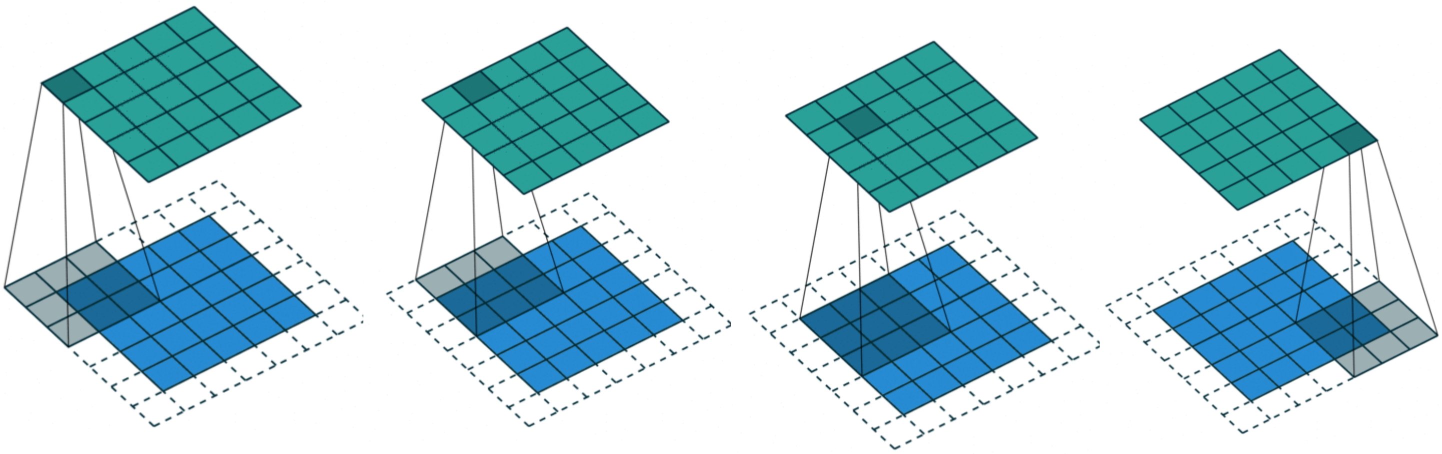
$h(x,y)$

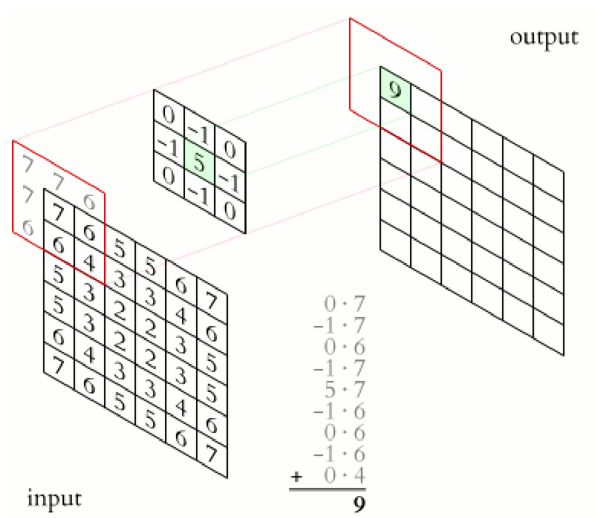
=

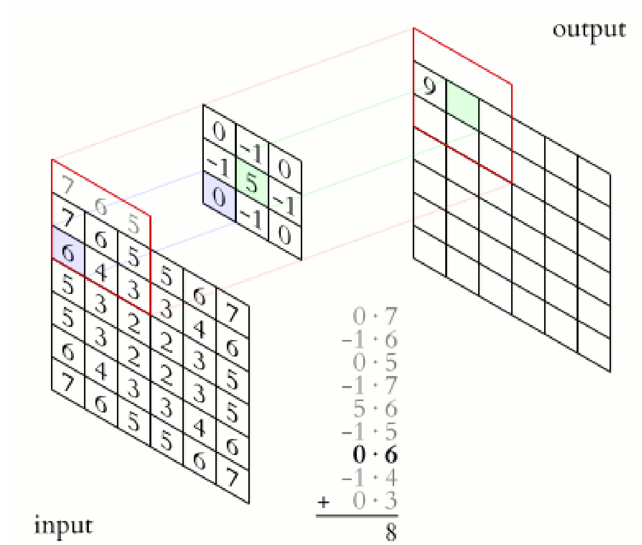
69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

$g(x,y)$

- Neighborhood filtering (convolution): The image on the left is convolved with the filter in the middle to yield the image on the right. The light blue pixels indicate the source neighborhood for the light green destination pixel.
- **Padding**
- Unpadded convolutions will crop away some of the borders if the kernel is larger than 1. (see previous example).
- To solve this issue and keeping the original size of the image, the image is padded with zeros before filtering or the image is extended 'replicate' by replicating the values in its outer border. (There are other ways).



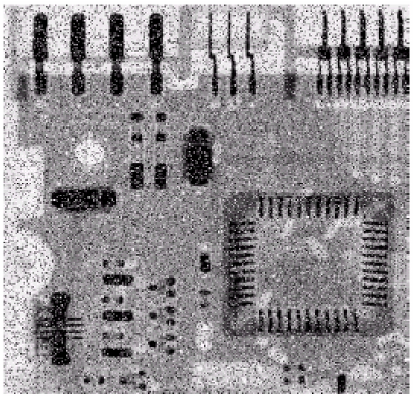




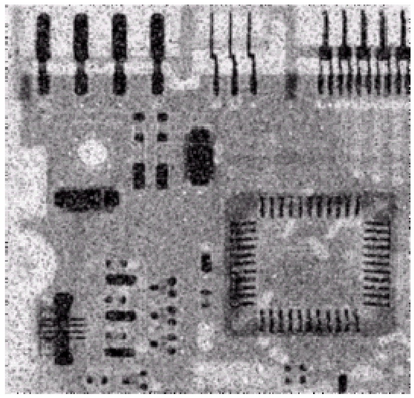
Order Statistical filters

- These refer to non-linear filters whose response is based on ordering the pixels contained in the neighborhood. Examples include **Maximum**, **Minimum**, **Median** and **Mode** filters.

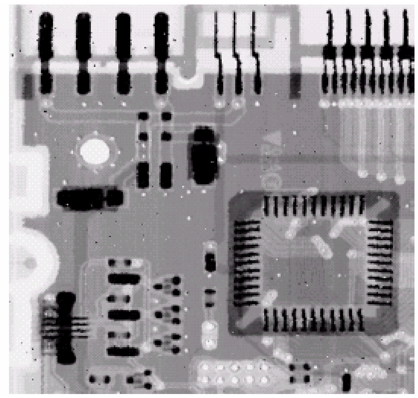
- The median which replaces the value at the centre by the median pixel value in the neighbourhood, (i.e. the middle element when they are sorted.
- Median filters are particularly useful in removing impulse noise, also known as salt-and-pepper.



Median filter



Noisy image



Averaging (3*3)

Median Filter Example

20	20	1	20	10	20	10	13	13
20	0	0	0	0	0	0	20	20
20	0	90	90	90	90	90	20	20
20	0	90	0	90	90	90	20	20
10	0	90	90	90	90	90	10	10
10	0	90	90	90	90	90	10	10
10	0	90	90	90	90	90	10	10
20	0	0	0	0	0	0	20	20
20	20	10	20	10	20	10	13	13

20	20	1	20	10	20	10	13	13
20	0	0	0	0	0	0	20	20
20	0	90	90	90	90	90	20	20
20	0	90	0	90	90	90	20	20
10	0	90	90	90	90	90	10	10
10	0	90	90	90	90	90	10	10
10	0	90	90	90	90	90	10	10
20	0	0	0	0	0	0	20	20
20	20	10	20	10	20	10	13	13

median(20 20 1 / 20 0 0 / 20 0 90)   median(0 0 0 1 20 20 20 20 90) = 20

average(0 0 0 1 20 20 20 20 90) = 19