

Image Filtering

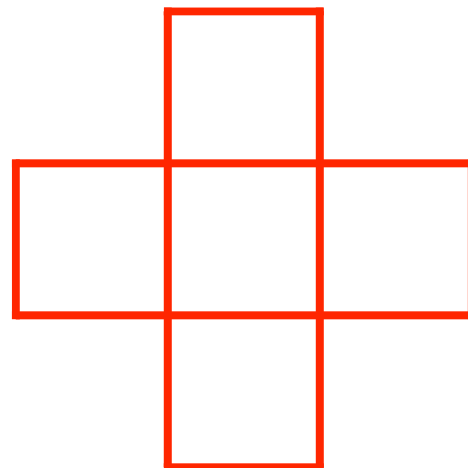
Lee Hwee Kuan
Bioinformatics Institute
A*STAR

The Sliding window - image filtering

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

13	16	18	10	18	28	26
21	37	25	26	16	35	33
27	28	35				

window



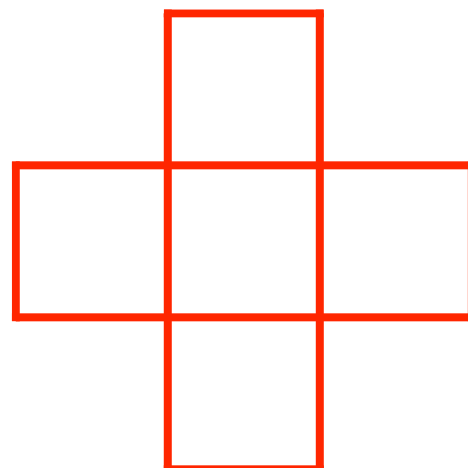
operations = sum of pixels

The Sliding window - image filtering

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

6	8	9	4	8	9	9
8	9	9	9	7	9	9
9	9	9				

window



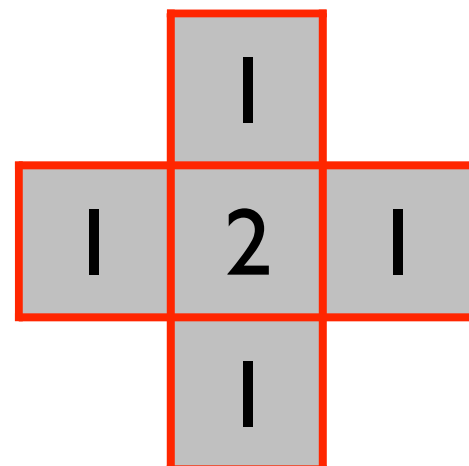
operations = max of pixels

The Sliding window - image filtering

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

9	9	5	5	8	7	3
11	10	9	8	12	10	4
6	5	8				

window



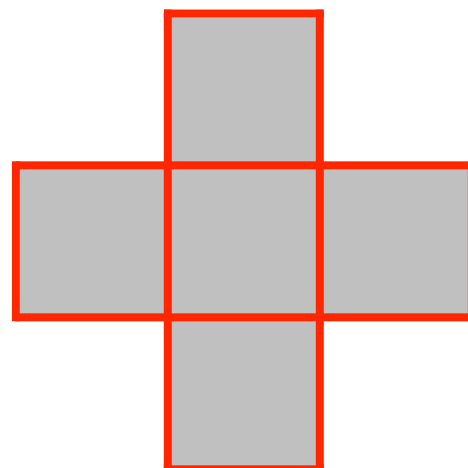
operations = weighted sum

Dilation as an image filter

	0	0	0	0	0	0	0
	0	0	1	0	0	0	0
	0	1	1	1	0	0	0
	0	0	1	1	0	0	0
	0	0	0	1	0	0	0
	0	0	0	1	0	0	0
	0	0	0	0	0	0	0

0	0	1	0	0	0	0
0	1	1	1	0	0	0
1	1	1	1	1	0	0
0	1	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0

window



operations = max pixel

What filter is use for erosion?

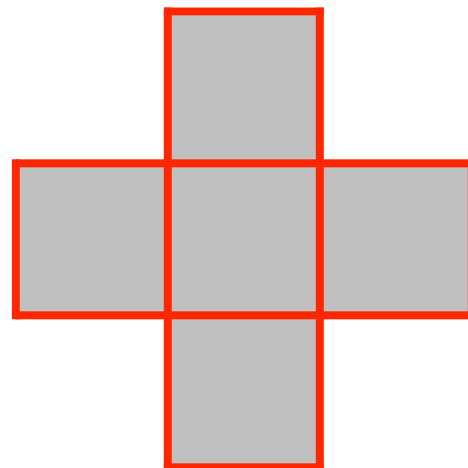
Mean filter

1 = white pixel, 0 = black pixel

0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

0	0	$\frac{1}{4}$	0	0	0	0
0	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	0	0	0
$\frac{1}{4}$	$\frac{2}{5}$	1	$\frac{3}{5}$	$\frac{1}{5}$	0	0
0	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	$\frac{1}{5}$	0	0
0	0	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{1}{5}$	0	0
0	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	0	0
0	0	0	$\frac{1}{5}$	0	0	0

window



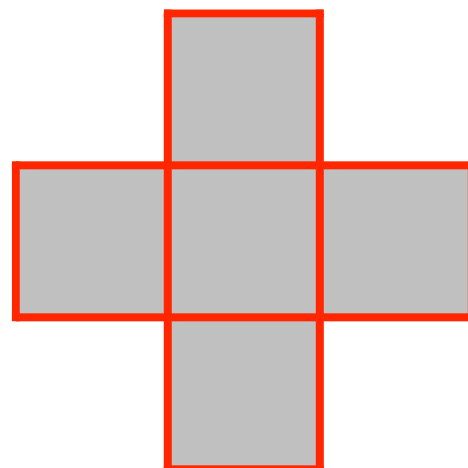
operations = mean

Mean filter

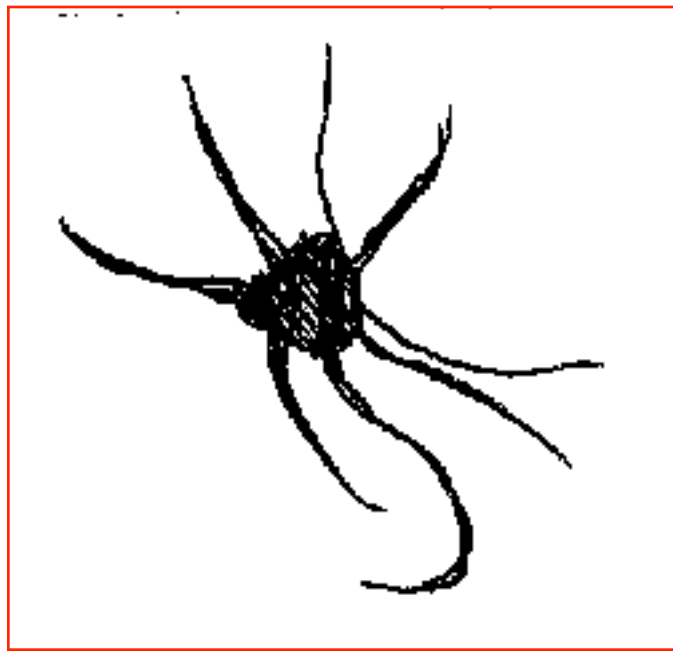
0	0	$1/4$	0	0	0	0
0	$2/5$	$2/5$	$2/5$	0	0	0
$1/4$	$2/5$	1	$3/5$	$1/5$	0	0
0	$2/5$	$2/5$	$4/5$	$1/5$	0	0
0	0	$2/5$	$3/5$	$1/5$	0	0
0	0	$1/5$	$2/5$	$1/5$	0	0
0	0	0	$1/5$	0	0	0

0	0.16	0.36	0.27	0	0	0
0.27	0.24	0.49	0.28	0.12	0	0
0.36	0.49	0.52	0.6	0.2	0.04	0
0.15	0.24	0.4	0.52	0.28	0.04	0
0	0.16	0.32	0.48	0.24	0.04	0
0	0.04	0.2	0.32	0.16	0.04	0
0	0	0.1	0.15	0.1	0	0

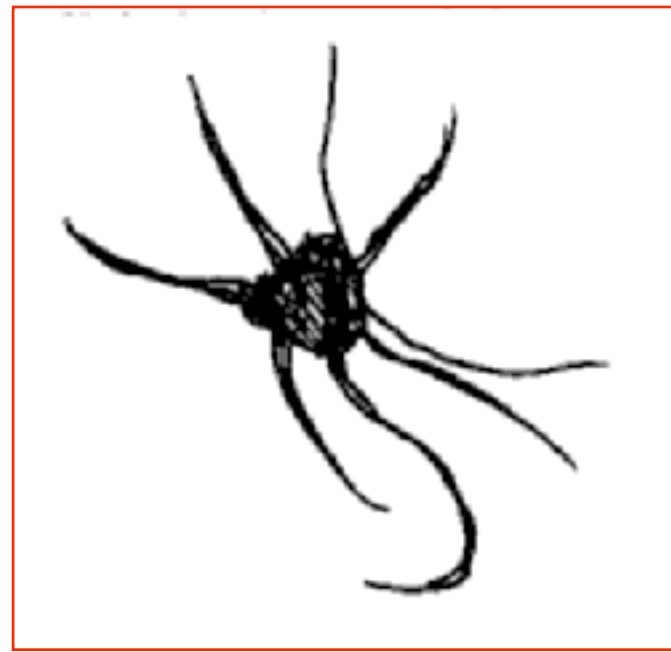
window



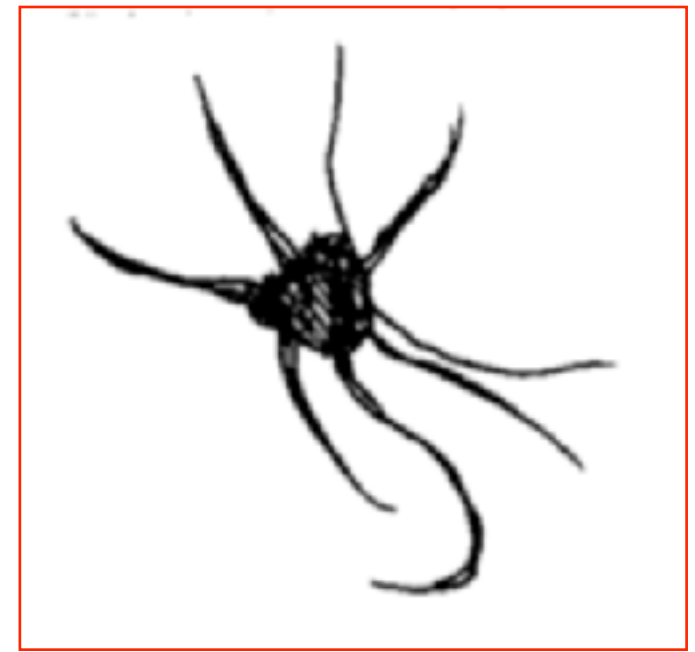
operations = mean



original image
344 x 332 pixels



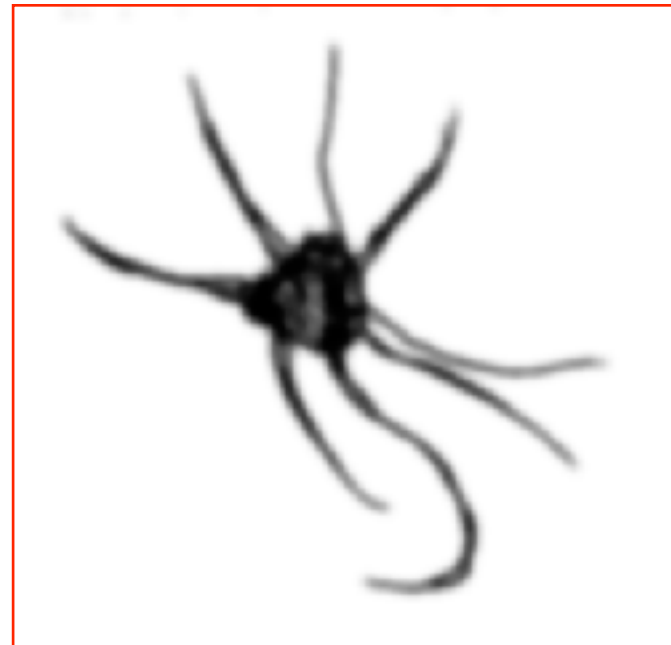
mean 1 time



mean 2 times



mean 3 times



mean 10 times



mean 1 time with radius 20

The Sliding window - image filtering

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

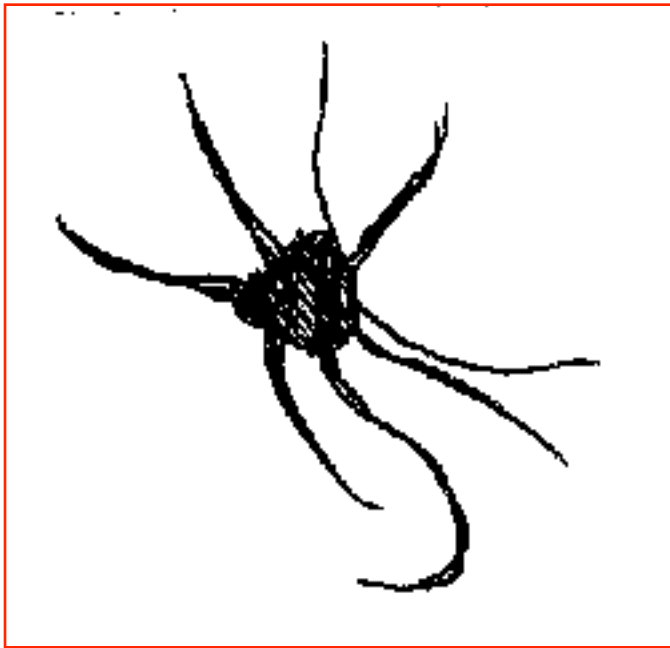
2	2	1	1	1	1	1
2	2	2	1	2	2	1
1	1	1				

window

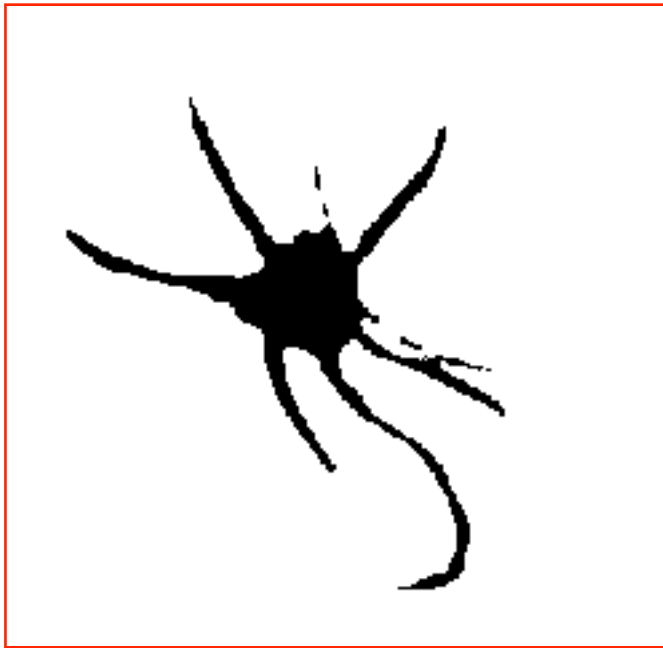
		1		
1		2		1
		1		

operations = median filter

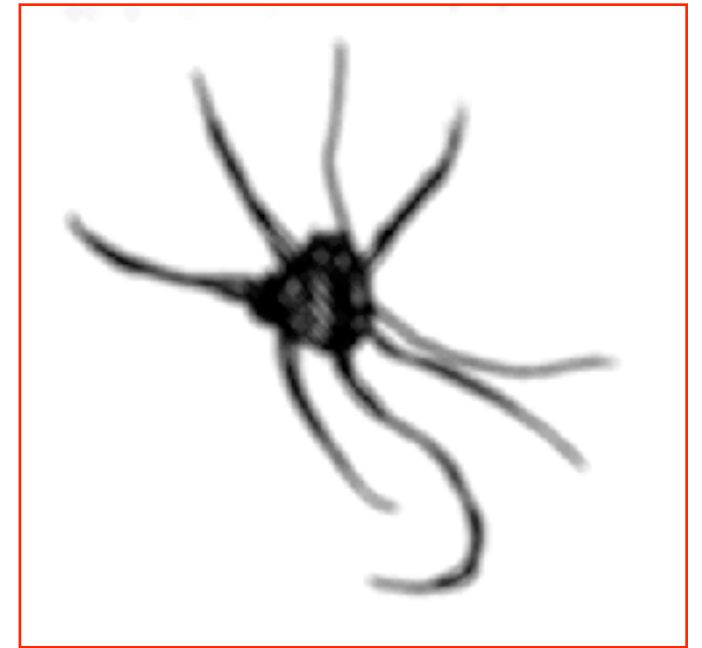
Median filter



original image
344 x 332 pixels



median filter
radius 4



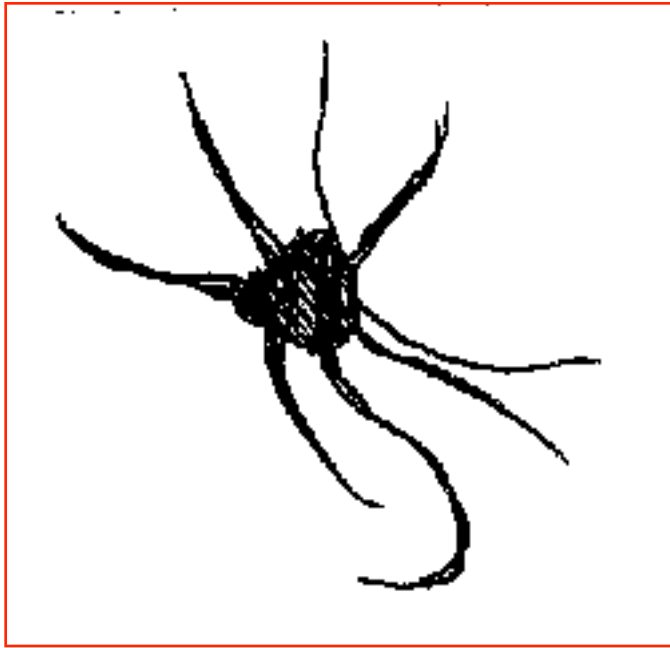
mean filter
radius 4

Gaussian filter

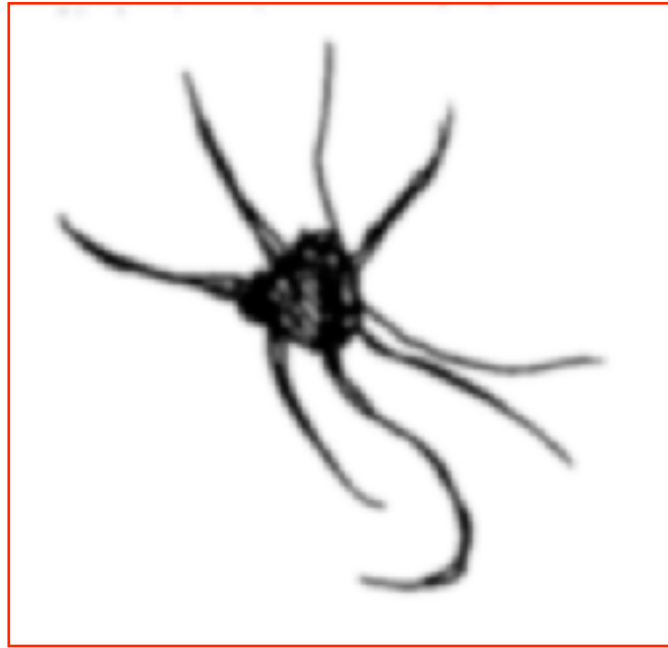
A weighted window

0.0075	0.01	0.02	0.01	0.0075
0.01	0.04	0.1	0.04	0.01
0.02	0.1	0.25	0.1	0.02
0.01	0.04	0.1	0.04	0.01
0.0075	0.01	0.02	0.01	0.0057

Put in weights with a
gaussian profile -> a
Gaussian filter



original image
344 x 332 pixels



filtered with sigma = 2
pixels



filtered with sigma = 8
pixels

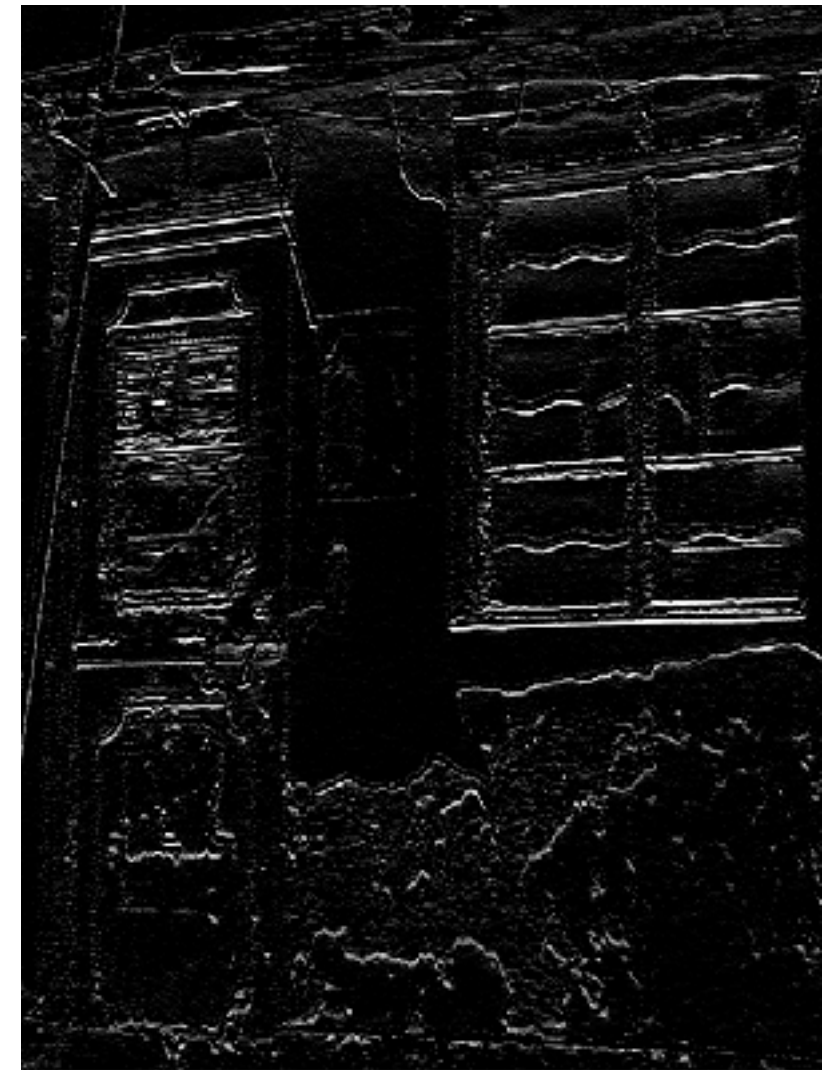
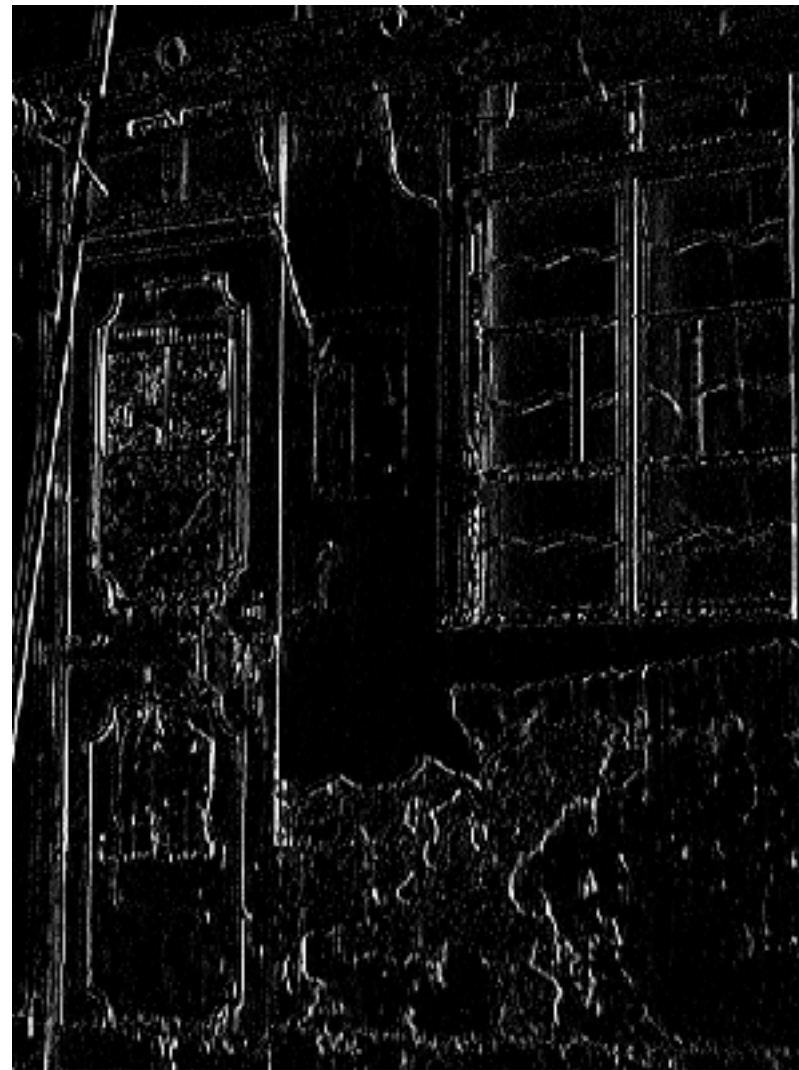
Edge filter

-1	0	1
-1	0	1
-1	0	1

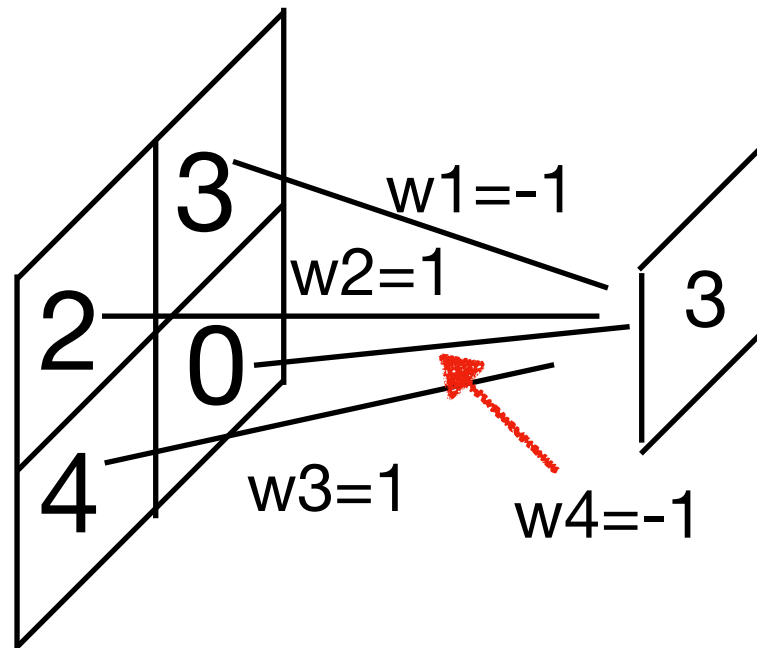
[illegible][illegible]

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1



Convolutional Neural Networks



$$3 = 3*(-1) + 2*(1) + 4*(1) + 0*(-1)$$

<https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d>

<https://deeplearning4j.org/convolutionalnetwork>

Convolutional Neural Networks

Simple example

<https://de.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>

State of the art

https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5