Practical content and lectures

Doctoral Training Centre (MPLS) Resources /Modules/ 2016-17 / Michaelmas term / Week 9-10: Foundations of Image Analysis / day01_practicals

Summary of course: Course-schedule-overview.html

Practicals: day01_practicals/day01-prac-overview.html

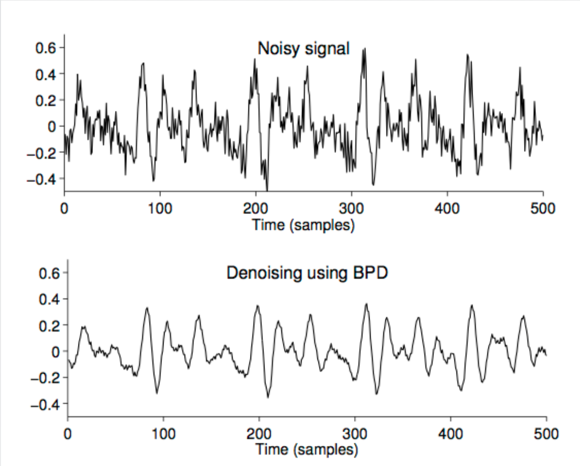
Source:

Image Processing

by Dominic Waithe
Weatherall Institute of Molecular Medicine

- Introduction to Image Processing
- Recap image formation
- Linear and non-linear filters
- Edge filters
- Filter banks
- Applied examples
- Deconvolution
- Conclusion

What is image processing



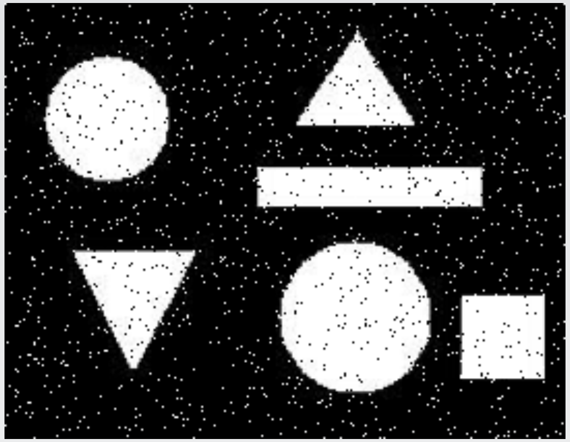


Image Processing usually involves a 2-D or 3-D images but has its roots in Signal Processing.

Signal processing is the science of transferring information contained in many different physical, symbolic, or abstract formats.

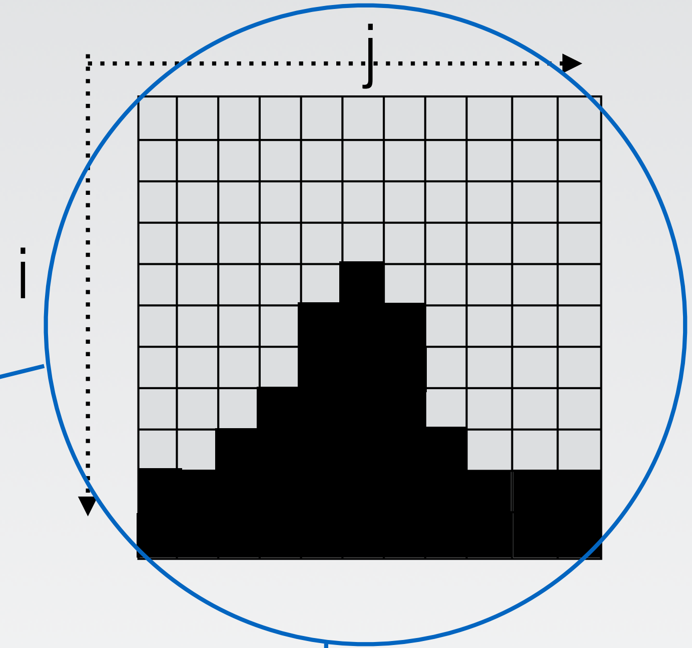
Usually we are trying to extract some type of meaning from our data, we process it to declutter our signal or focus on a particular aspect.

Source: http://eeweb.poly.edu/iselesni/lecture_notes/sparsity_intro/bpd.png
http://photos1.blogger.com/blogger/82/1614/1600/fig1.2.jpg

Image representation recap: Binary Image

Computer images are 2D arrays of numbers:

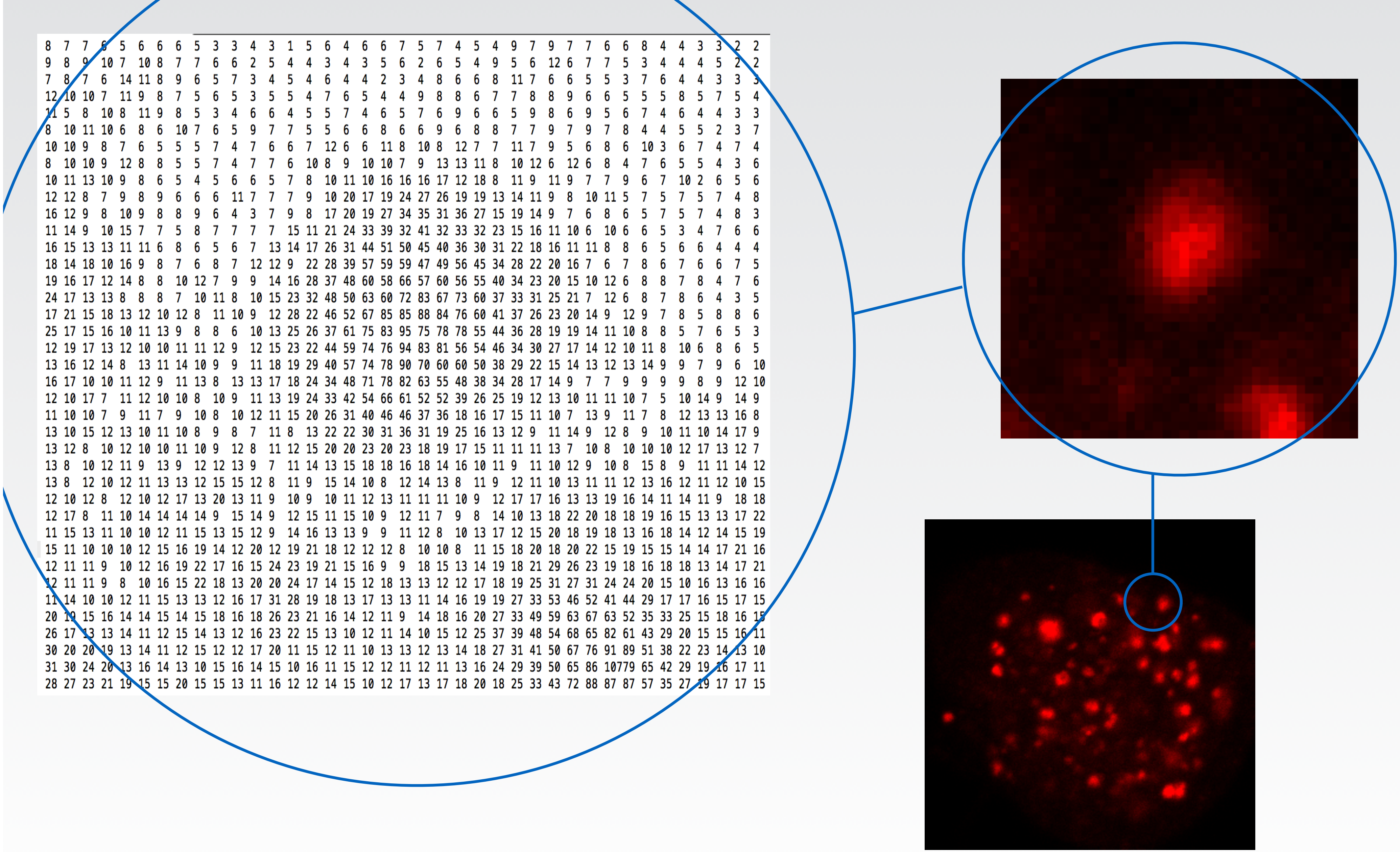
```
0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,1,0,0,0,0,0
0,0,0,0,1,1,1,0,0,0,0
0,0,0,0,1,1,1,0,0,0,0
0,0,0,0,1,1,1,0,0,0,0
0,0,0,1,1,1,1,0,0,0,0
0,0,1,1,1,1,1,1,0,0,0
1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,1,1,1
```



Source: http://www.uff.br/cdme/matrix/matrix-html/matrix_boolean/matrix_boolean_en.html

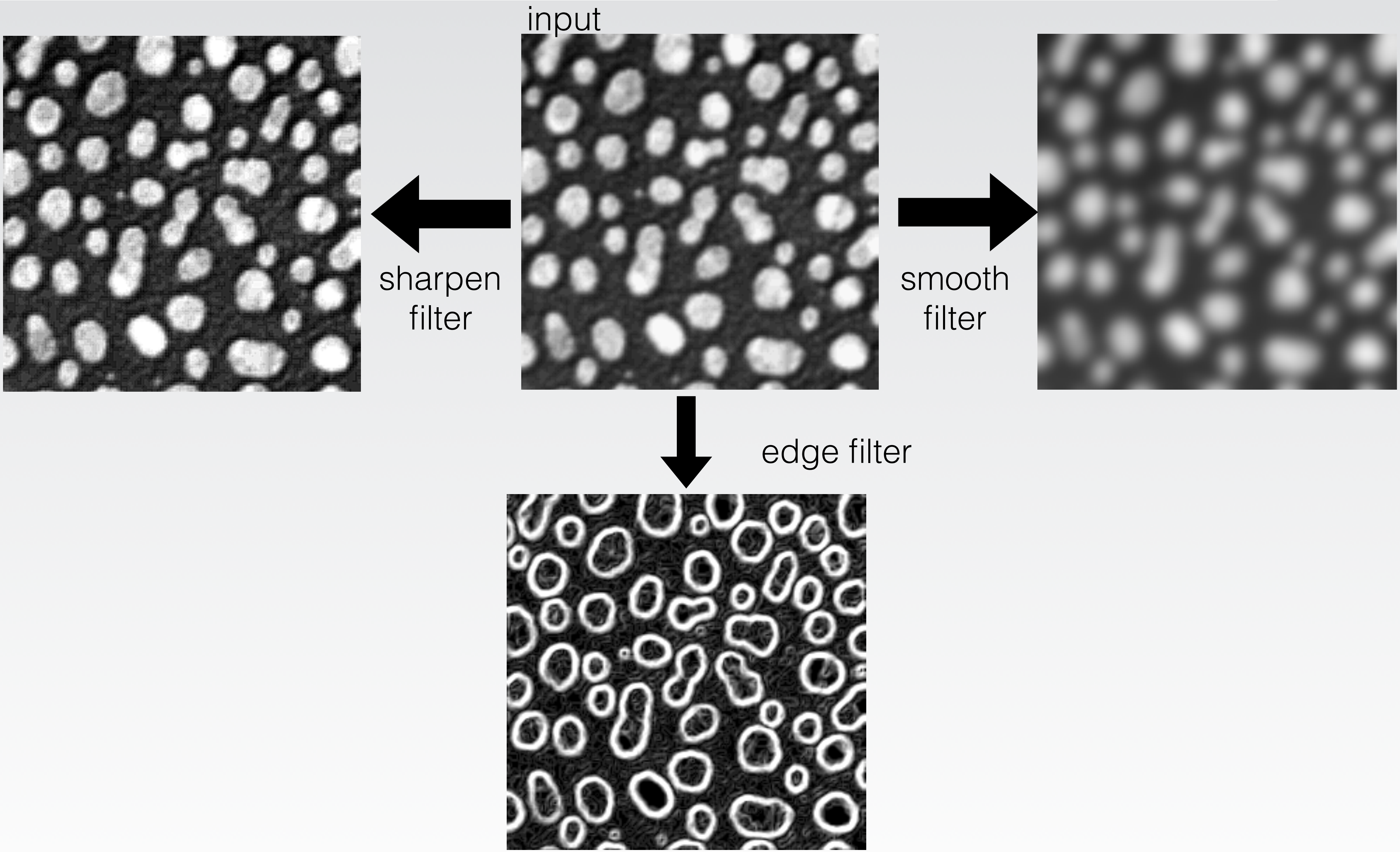
Image representation recap: Grayscale Image

Computer images are 2D arrays of numbers:



Source: http://www.uff.br/cdme/matrix/matrix-html/matrix_boolean/matrix_boolean_en.html

Image processing: Image Filtering



Linear and non-linear filtering.

Linear Filters	non-linear filters
mean filter	median filter
Gaussian filter	percentile filter
Sobel filter	maximum filter
Prewitt filter	minimum filter
Haar-like filter	
Gabor filter	
Difference of Gaussians	

Convolution - recap used for linear filtering

continuous convolution operator

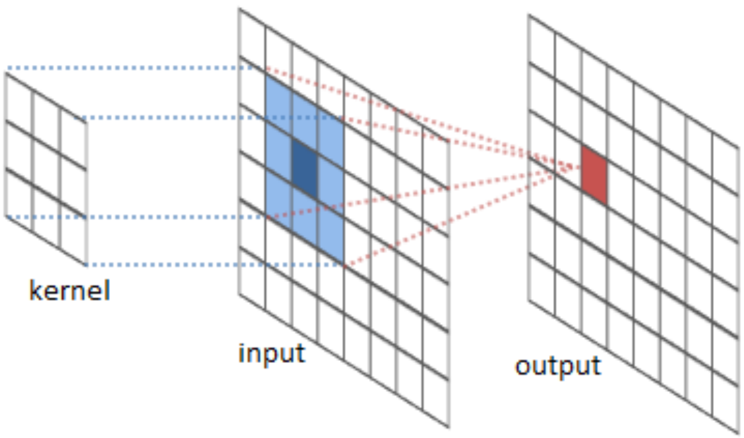
$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

discrete convolution operator

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

$$1/9 \times \boxed{\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}}$$

kernel: 3x3 mean filter

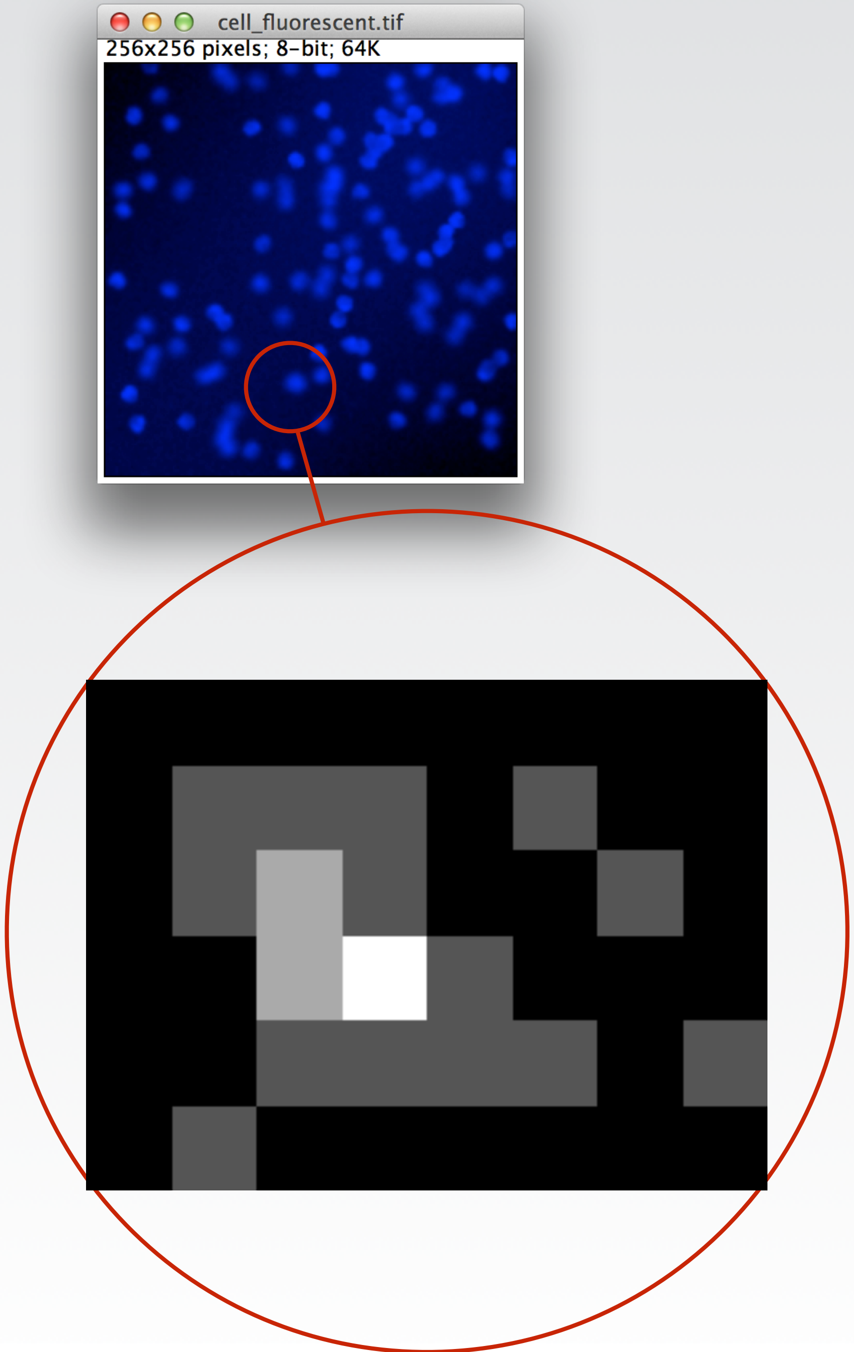


$(a_{-1-1} * b_{-1-1}) +$
$(a_{-1\,0} * b_{-1\,0}) +$
$(a_{-1\,1} * b_{-1\,1}) +$ $= \text{output}$
$(a_{0\,-1} * b_{0\,-1}) +$
$(a_{0\,0} * b_{0\,0}) +$

etc, etc

Convolution is the weighted sum of the pixel neighbourhood. The kernel defines the weights.

Linear filter: Mean filter



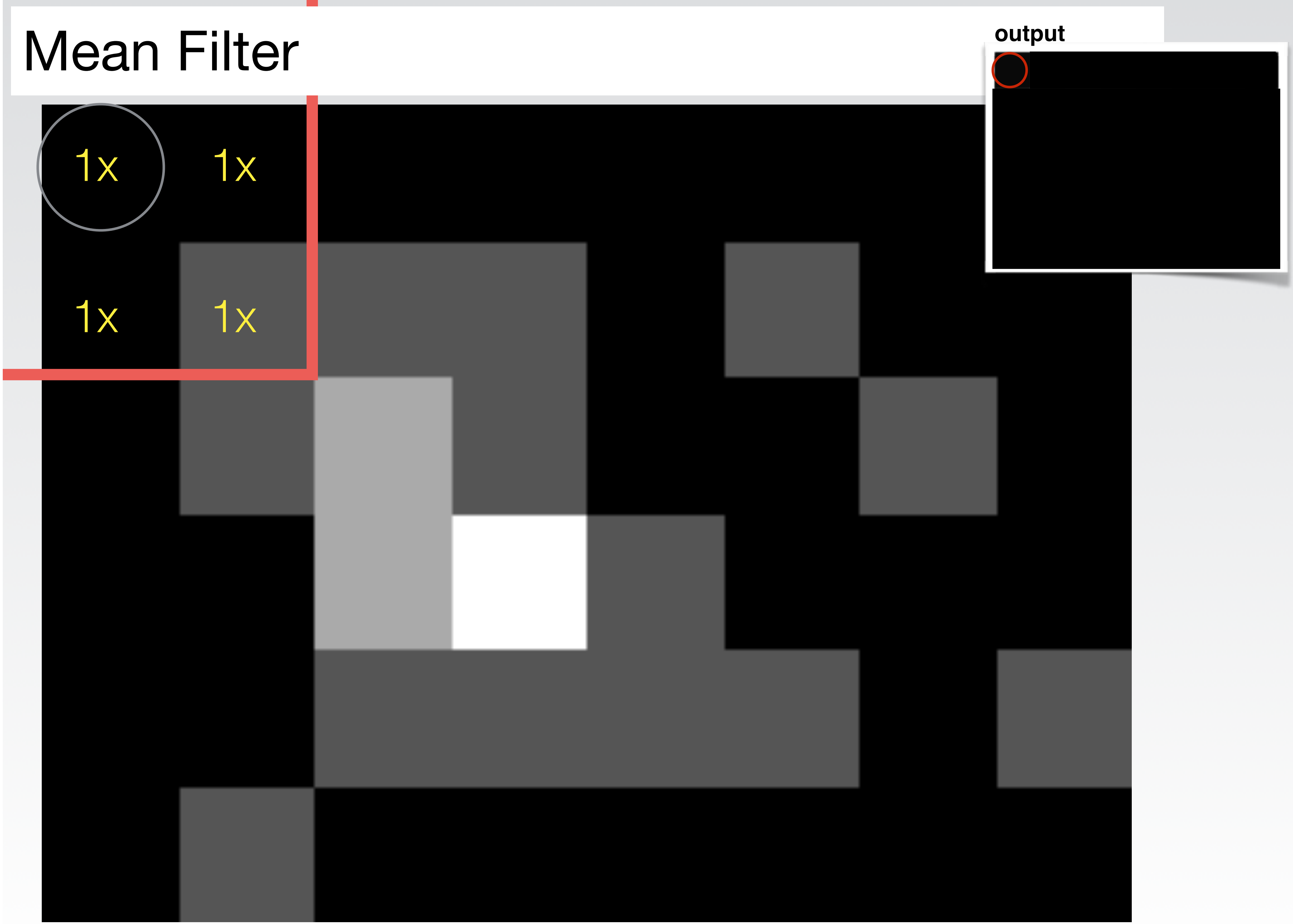
1	1	1
1	1	1
1	1	1

kernel: 3x3 mean filter

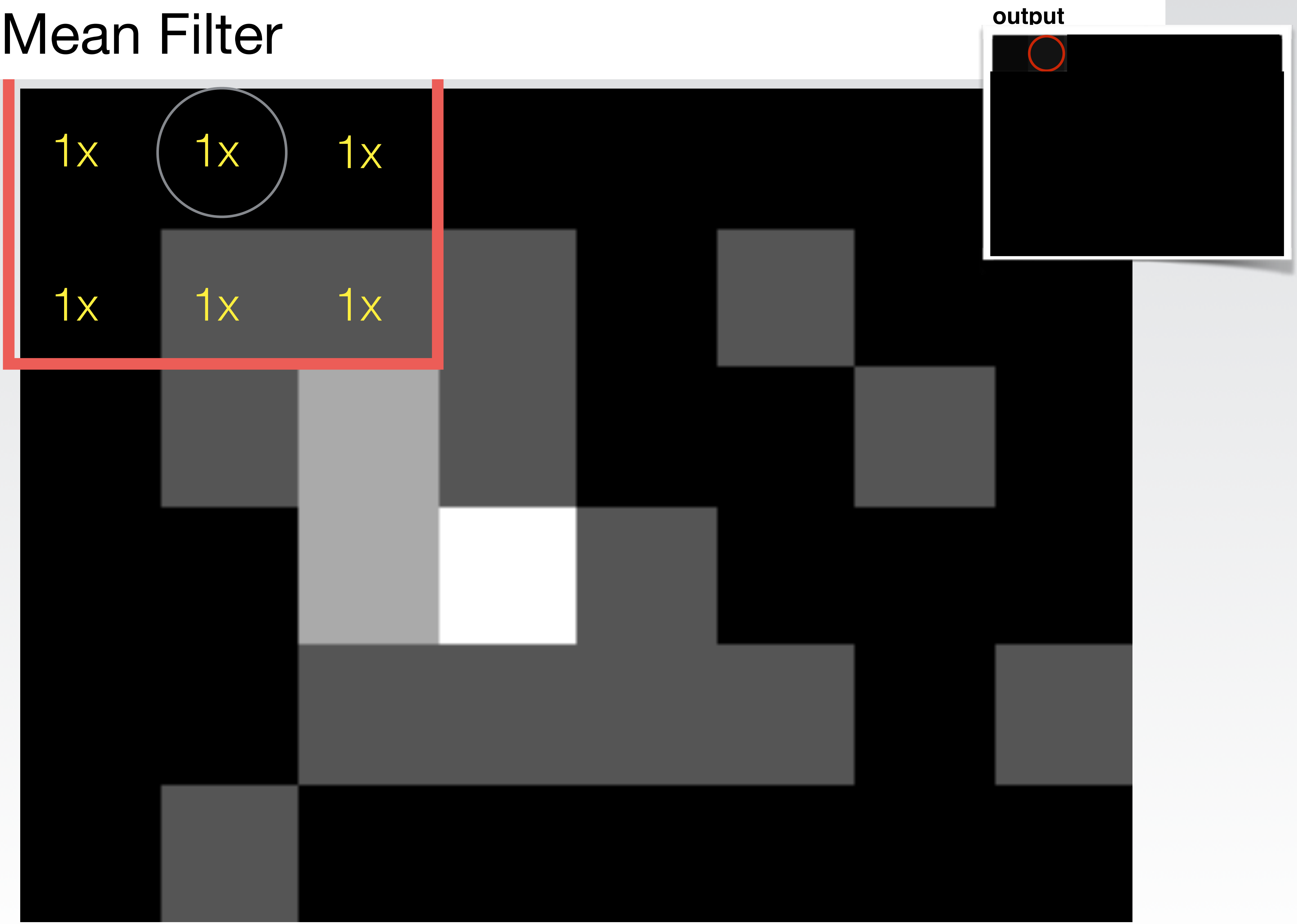
[[1, 1, 1, 1, 1, 1, 1, 1],
[1, 2, 2, 2, 1, 2, 1, 1],
[1, 2, 3, 2, 1, 1, 2, 1],
[1, 1, 3, 4, 2, 1, 1, 1],
[1, 1, 2, 2, 2, 2, 1, 2],
[1, 2, 1, 1, 1, 1, 1, 1]]

Image region

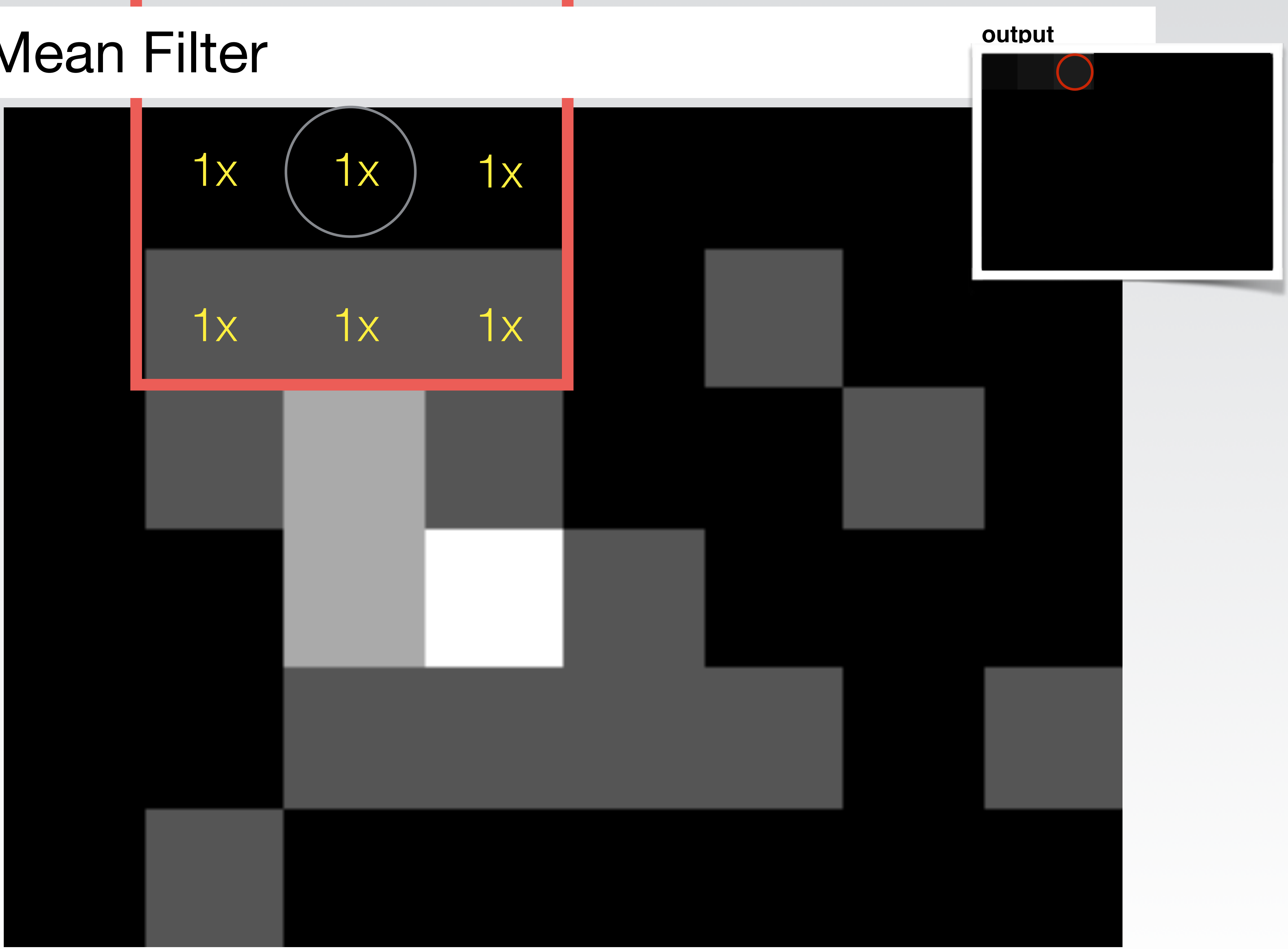
Mean Filter



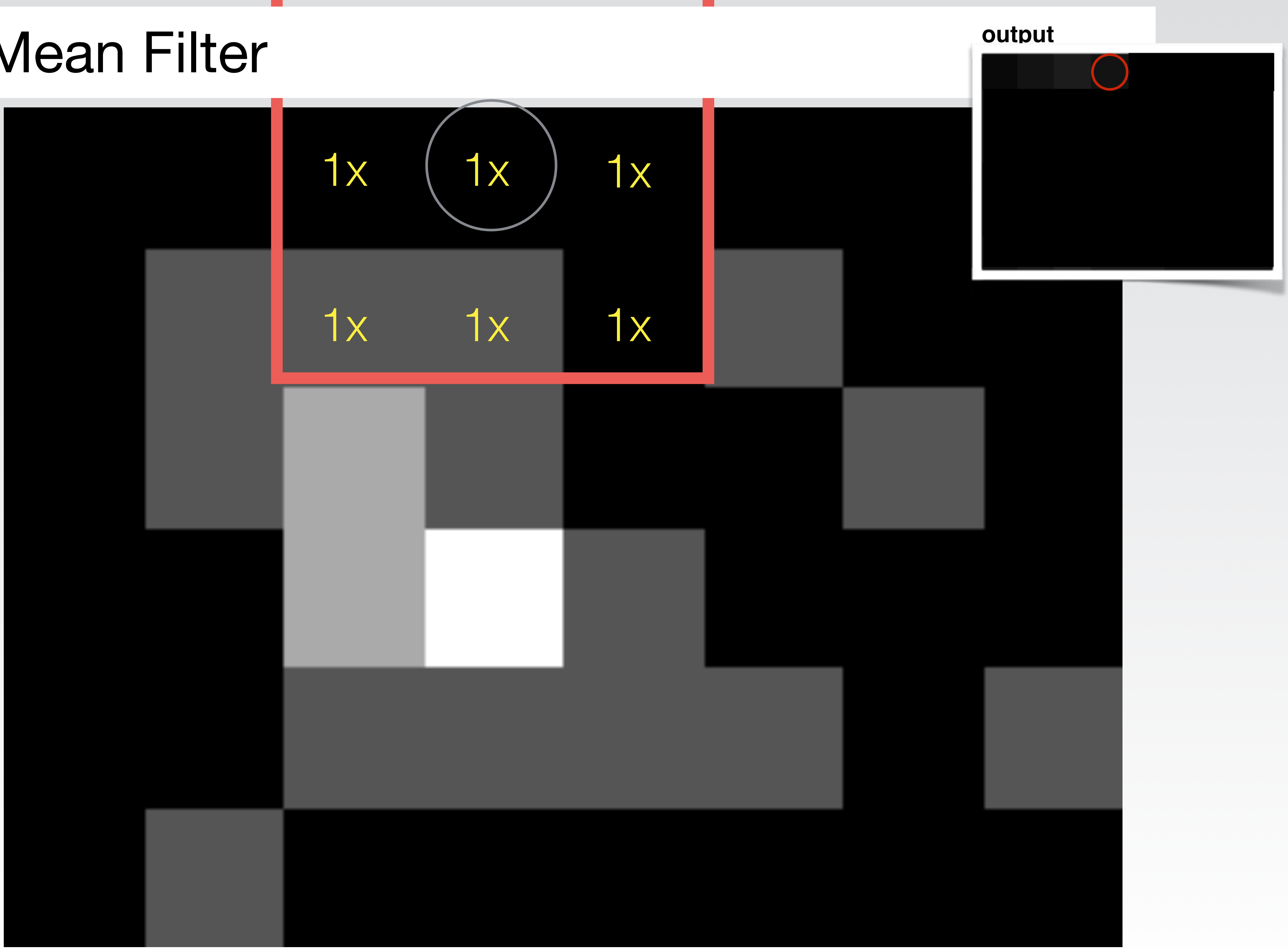
Mean Filter



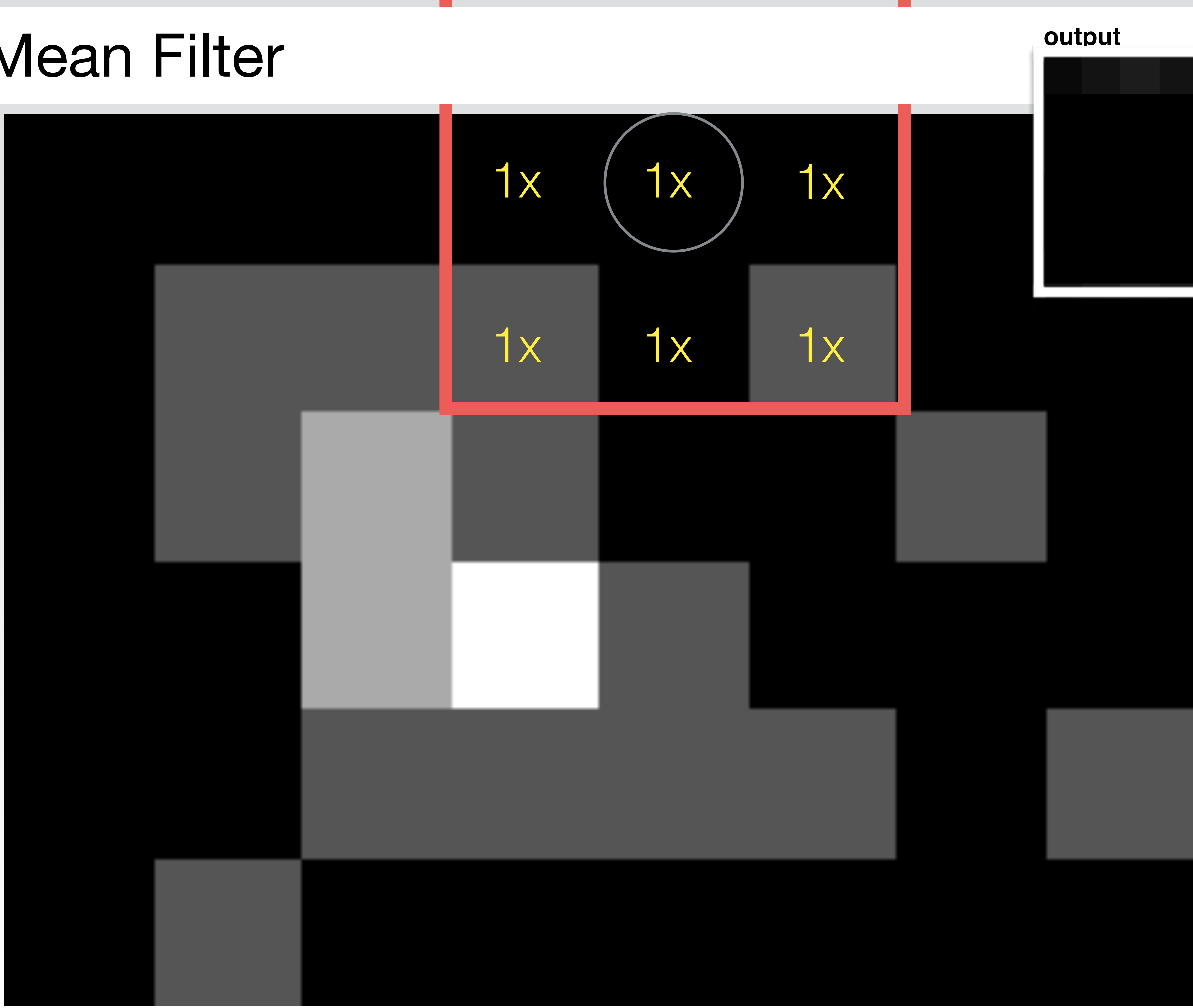
Mean Filter

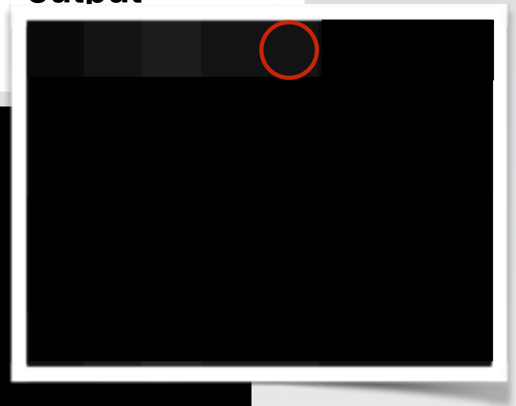


Mean Filter

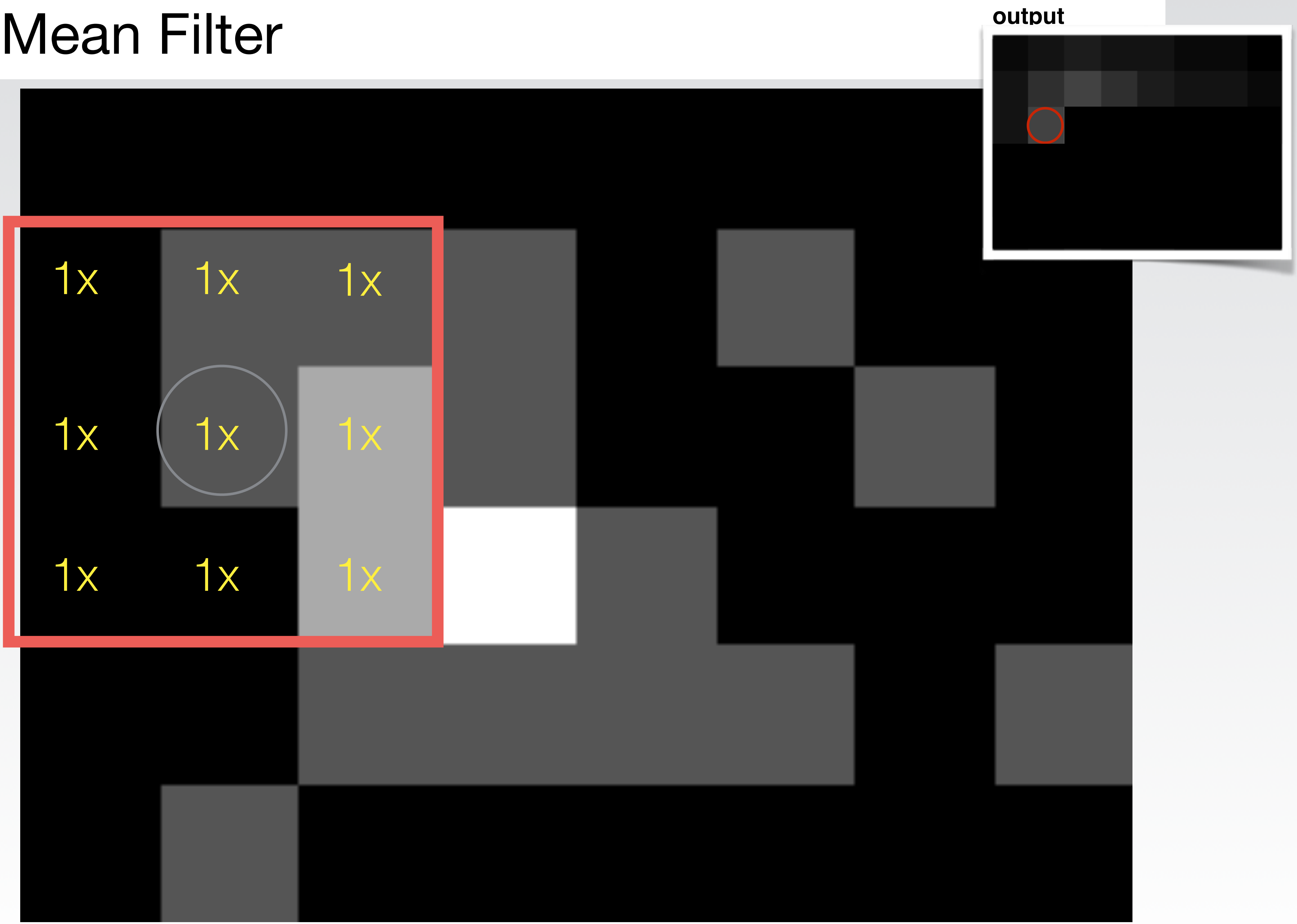


Mean Filter

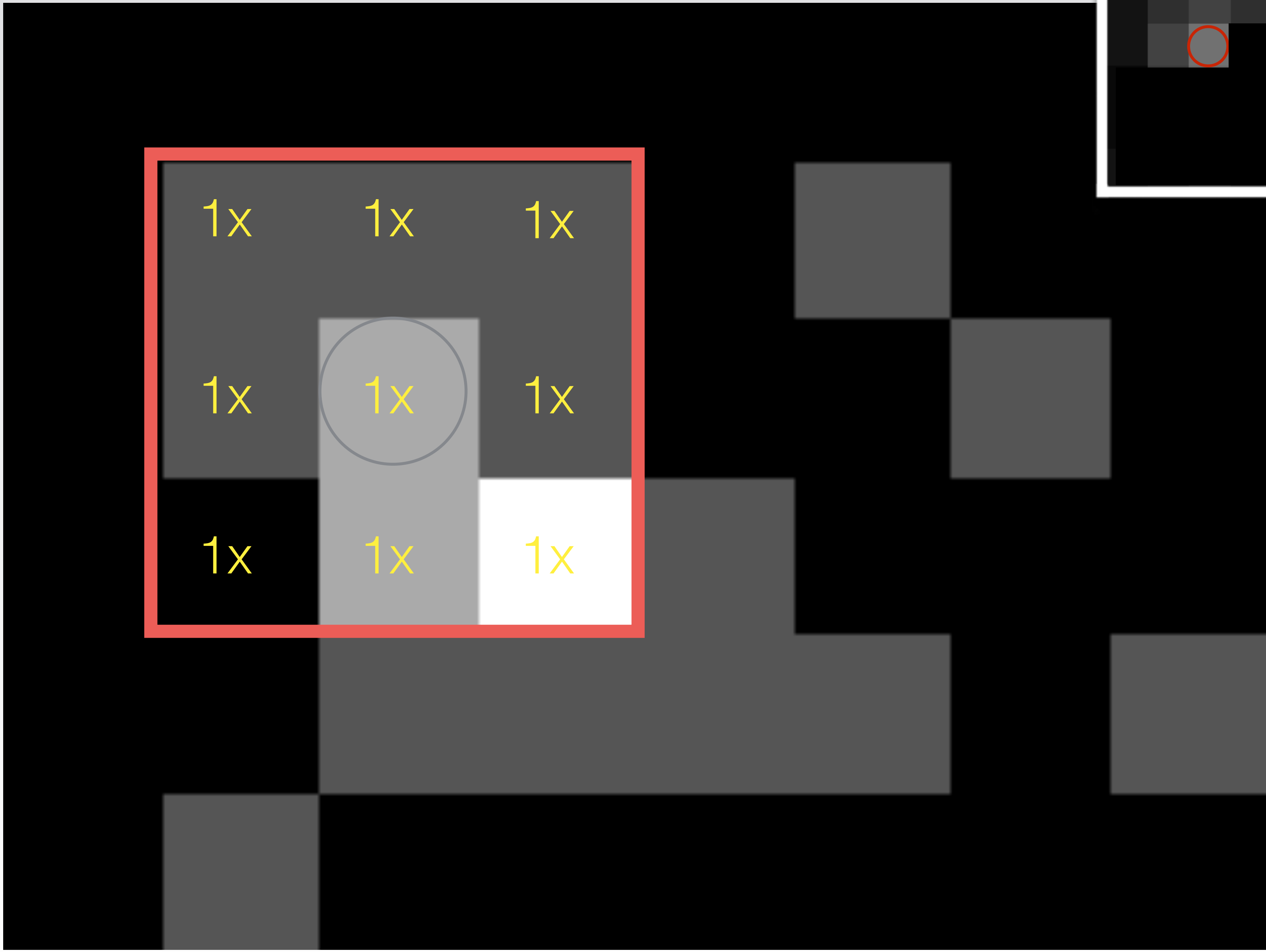


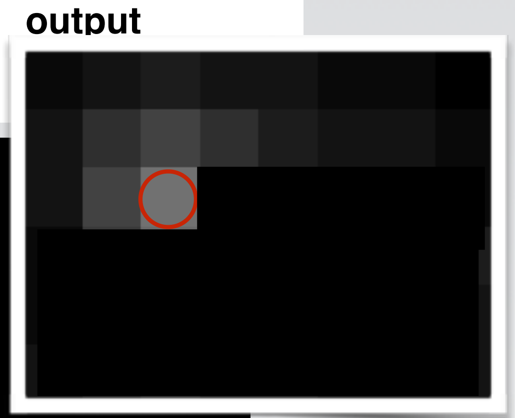


Mean Filter

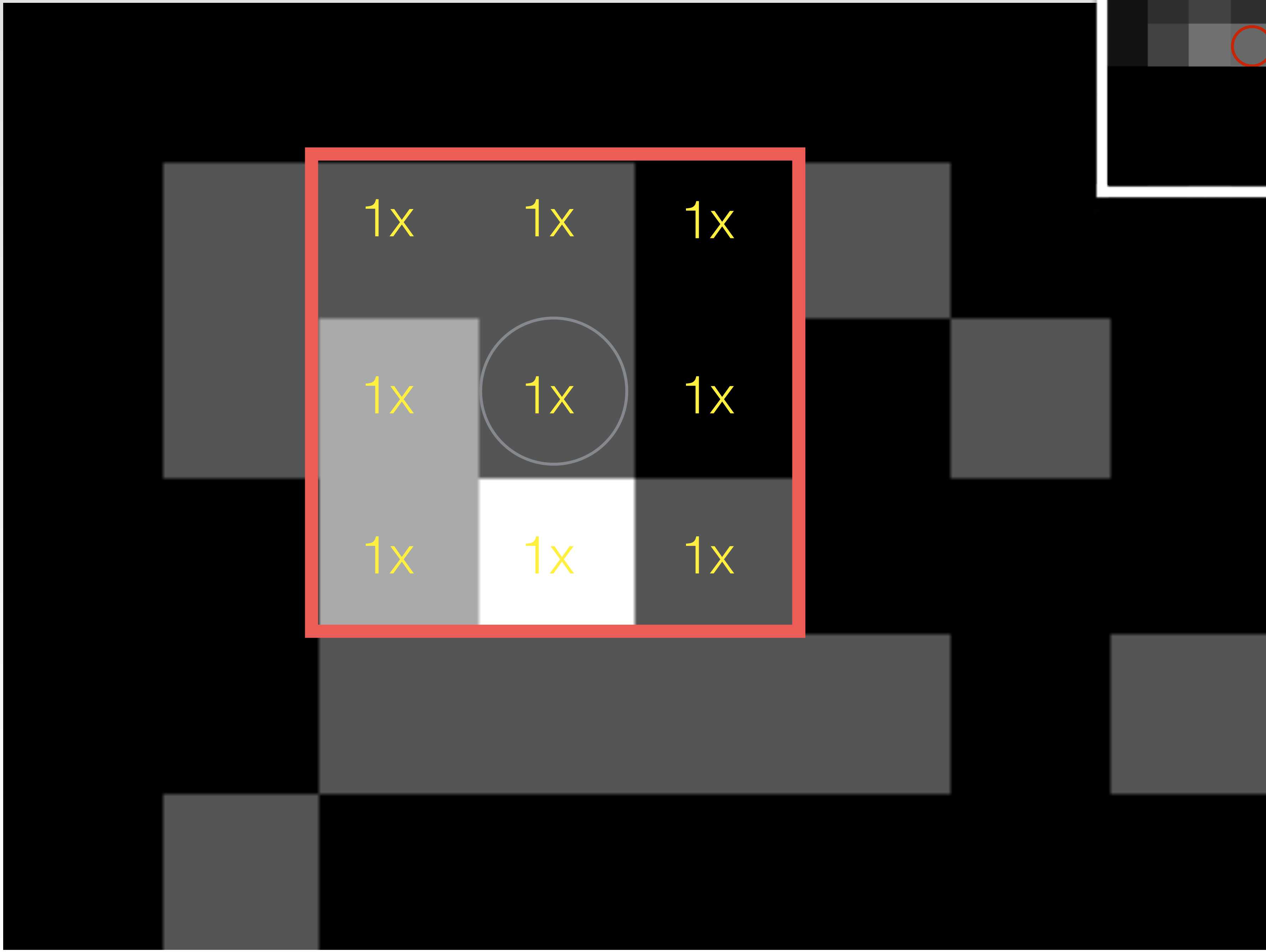


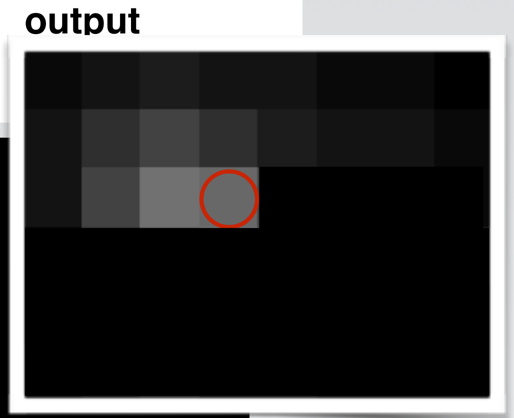
Mean Filter



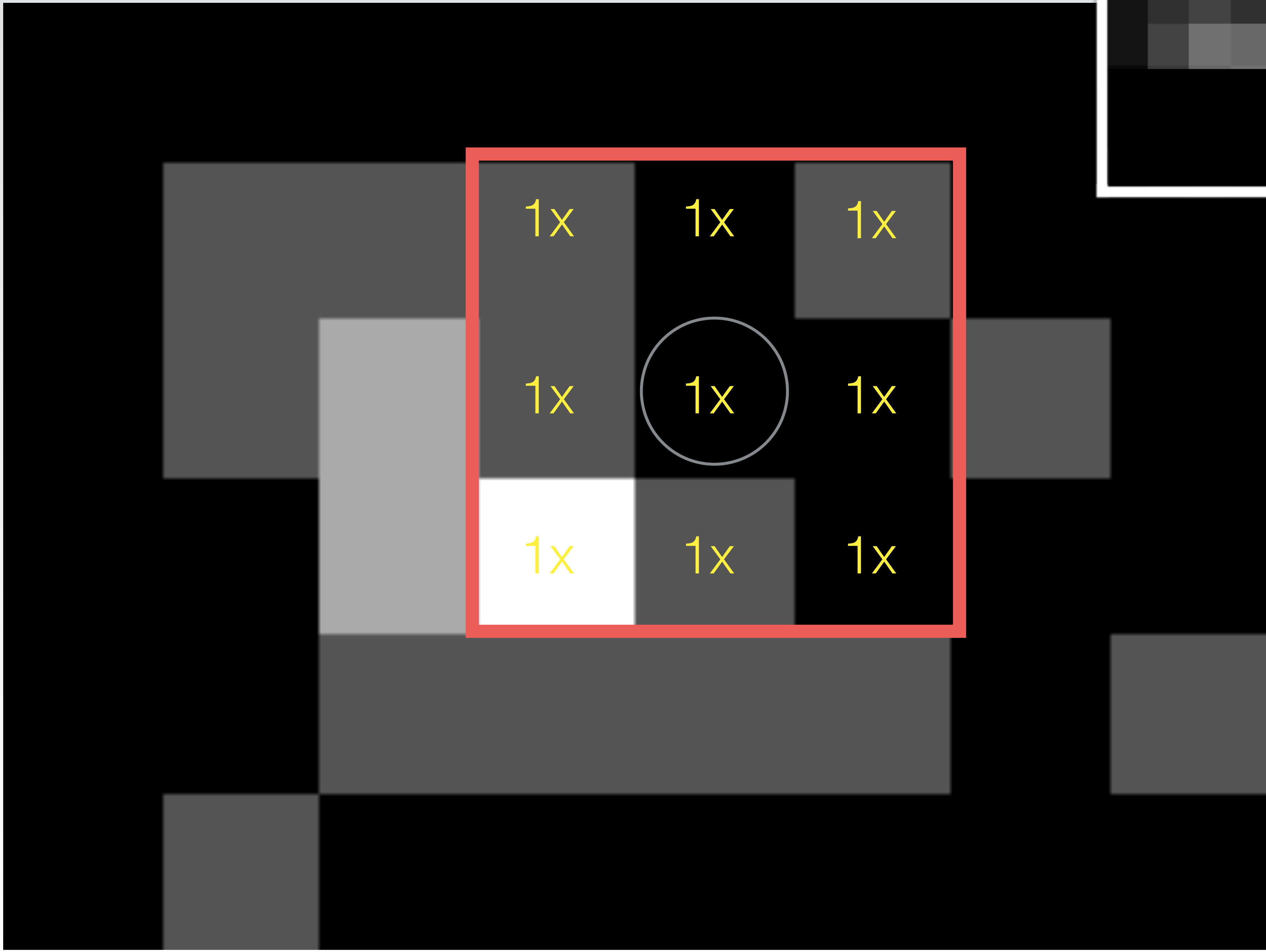


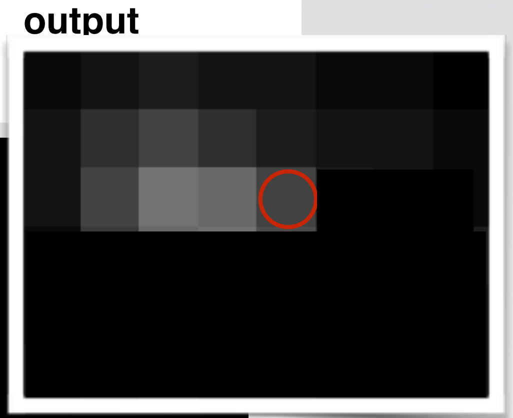
Mean Filter



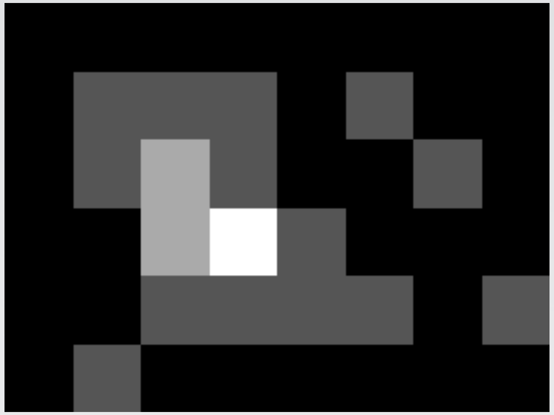


Mean Filter

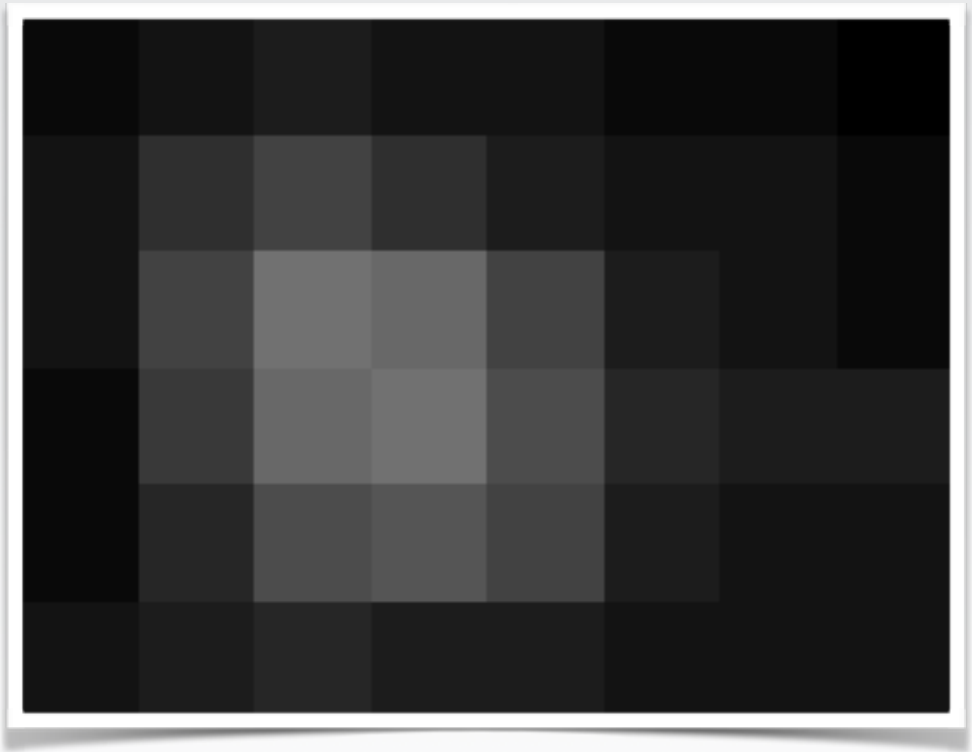




Result:

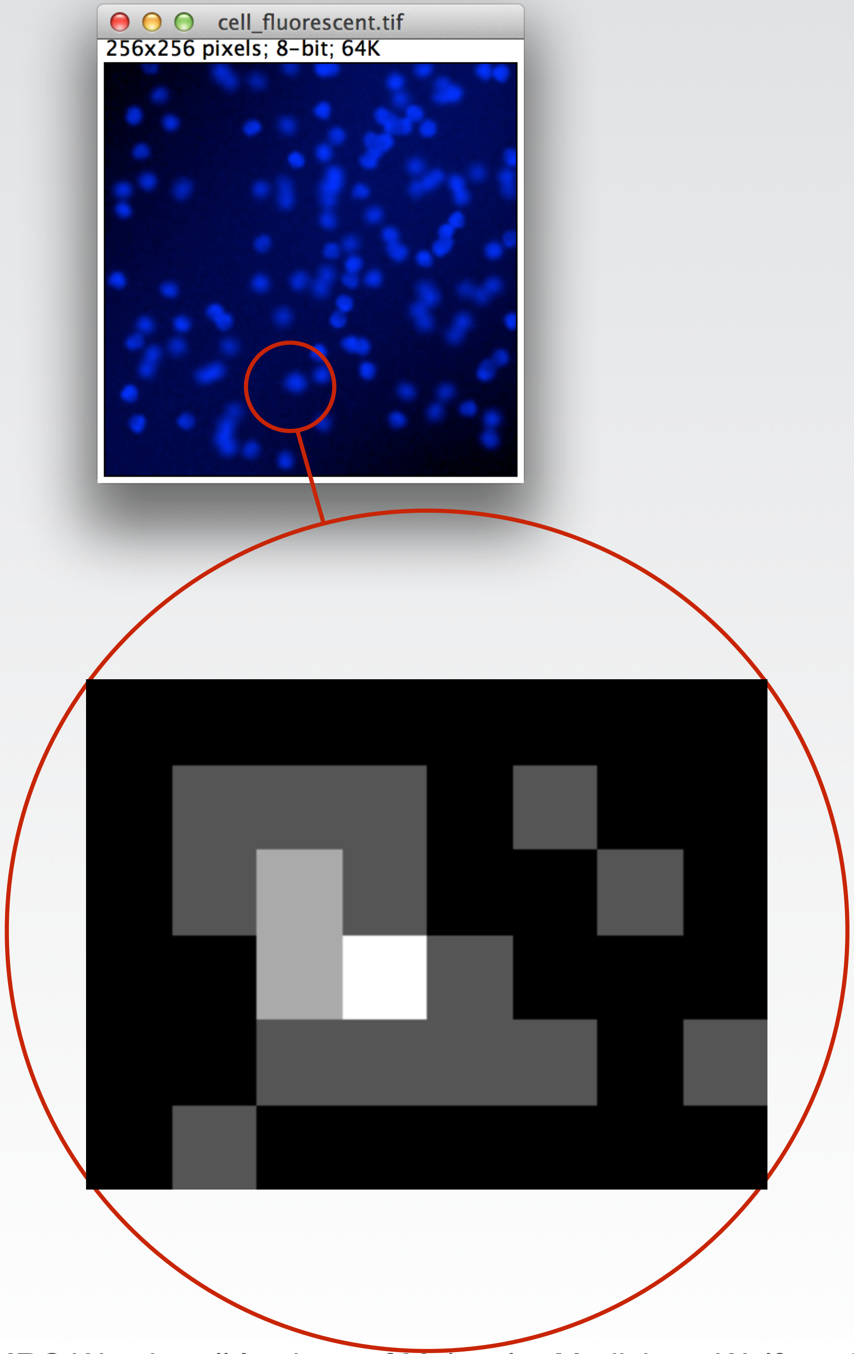


The output is a smoothed representation of the input.



Source:

Non-linear filter: Maximum filter

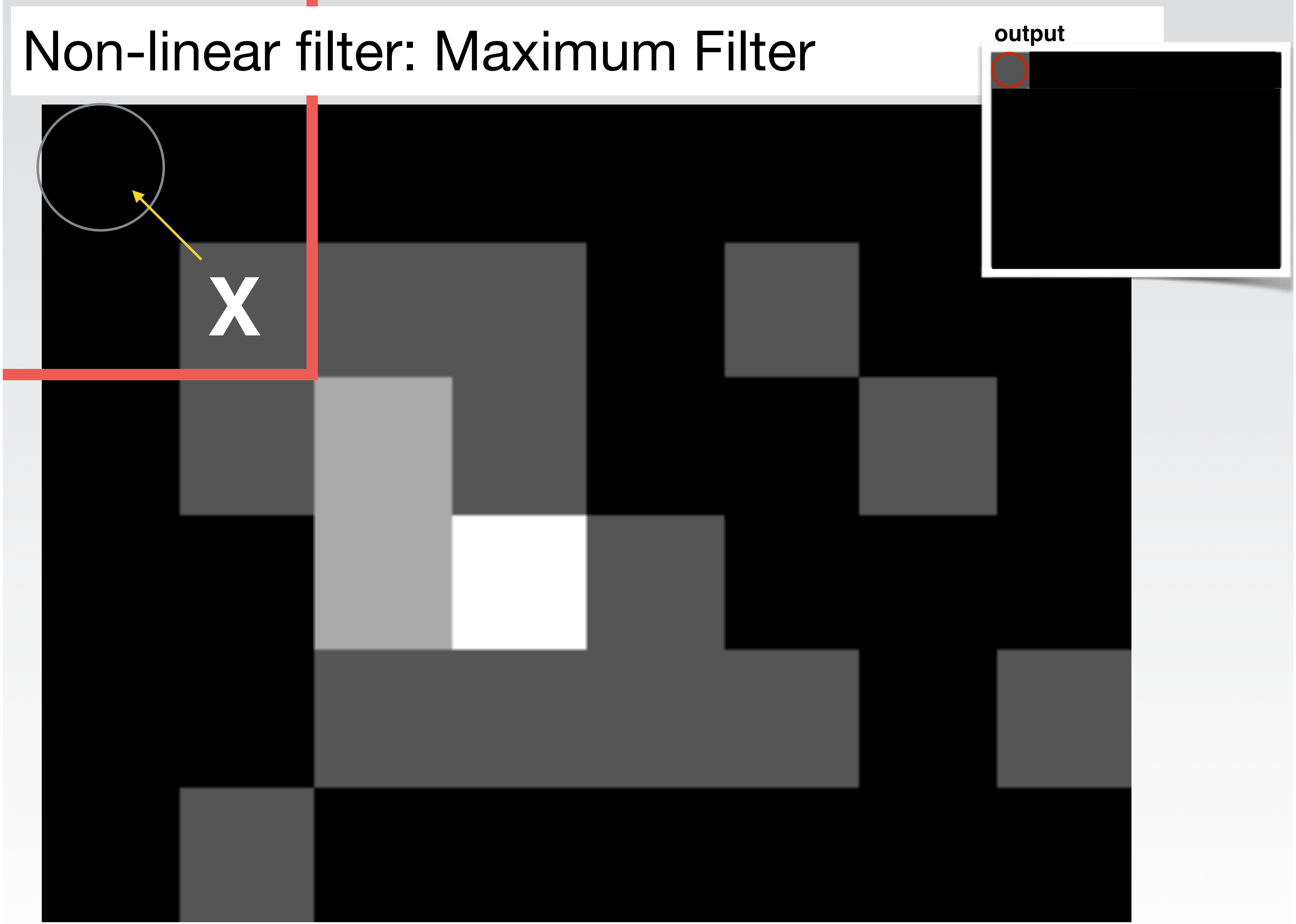


?

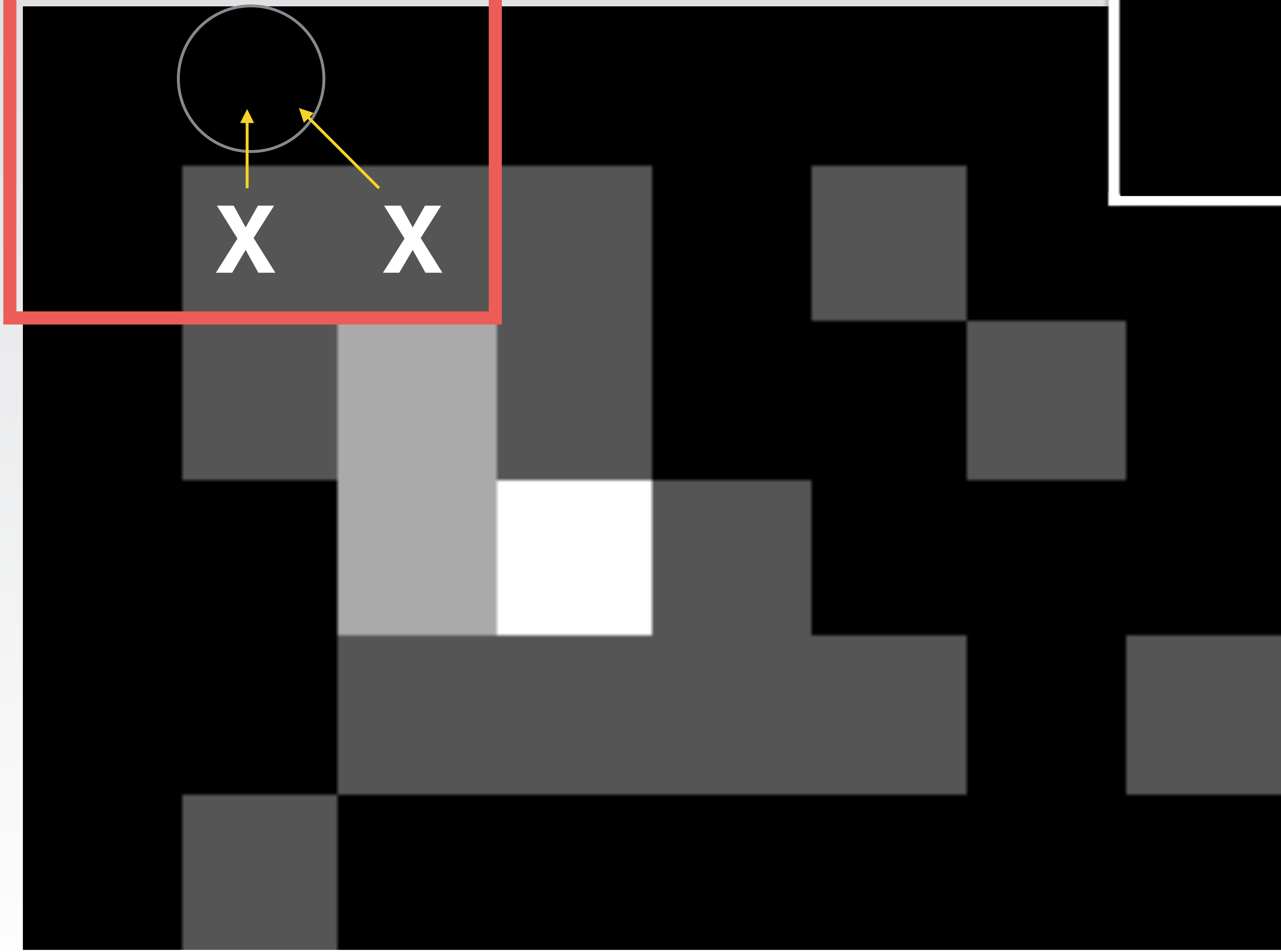
kernel can not be expressed as is non-linear

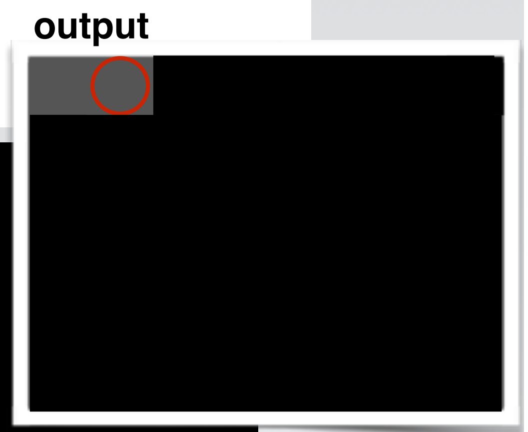
```
[[1, 1, 1, 1, 1, 1, 1, 1],
 [1, 2, 2, 2, 1, 2, 1, 1],
 [1, 2, 3, 2, 1, 1, 2, 1],
 [1, 1, 3, 4, 2, 1, 1, 1],
 [1, 1, 2, 2, 2, 2, 1, 2],
 [1, 2, 1, 1, 1, 1, 1, 1]]
```

Non-linear filter: Maximum Filter

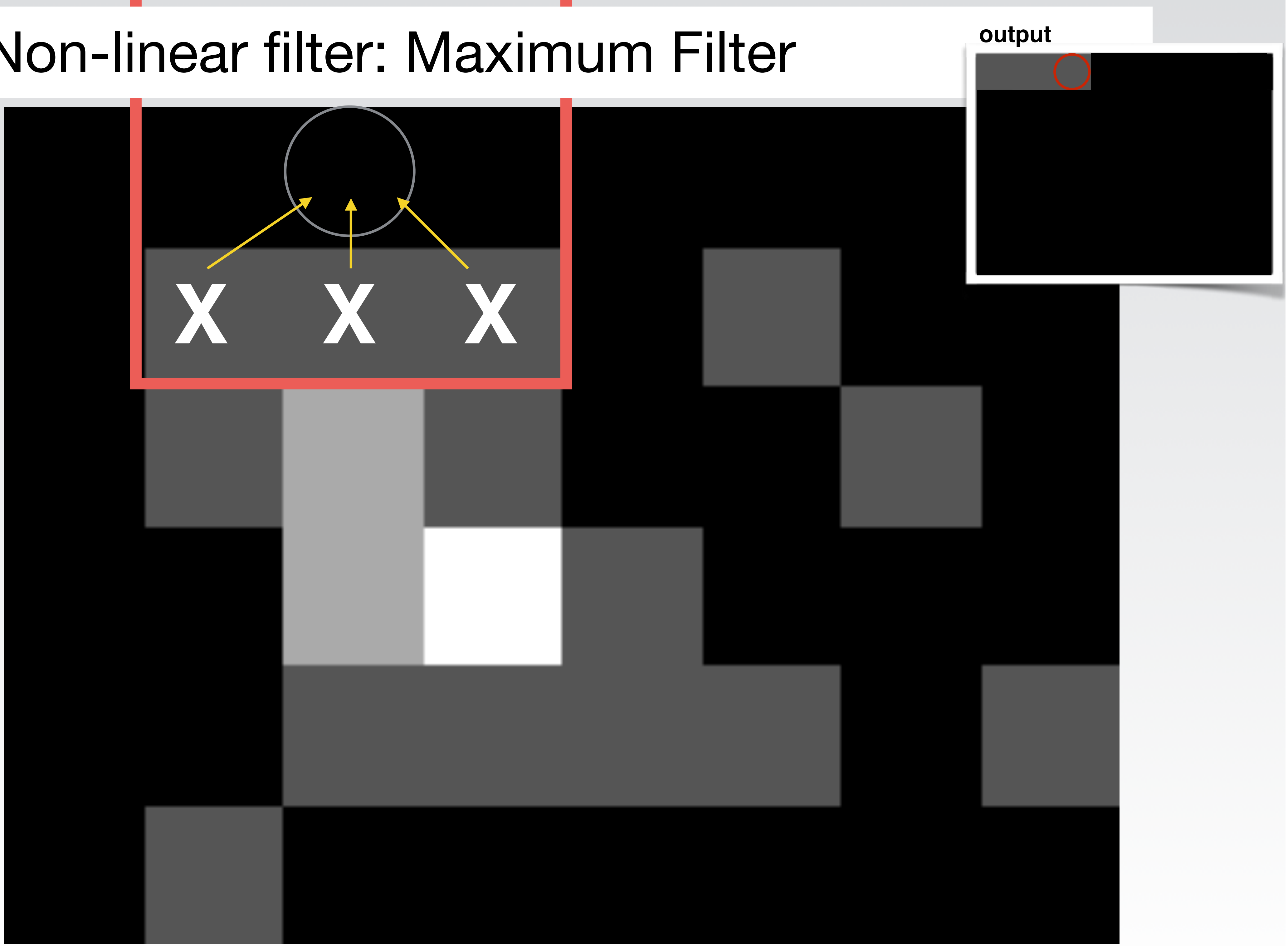


Non-linear filter: Maximum Filter

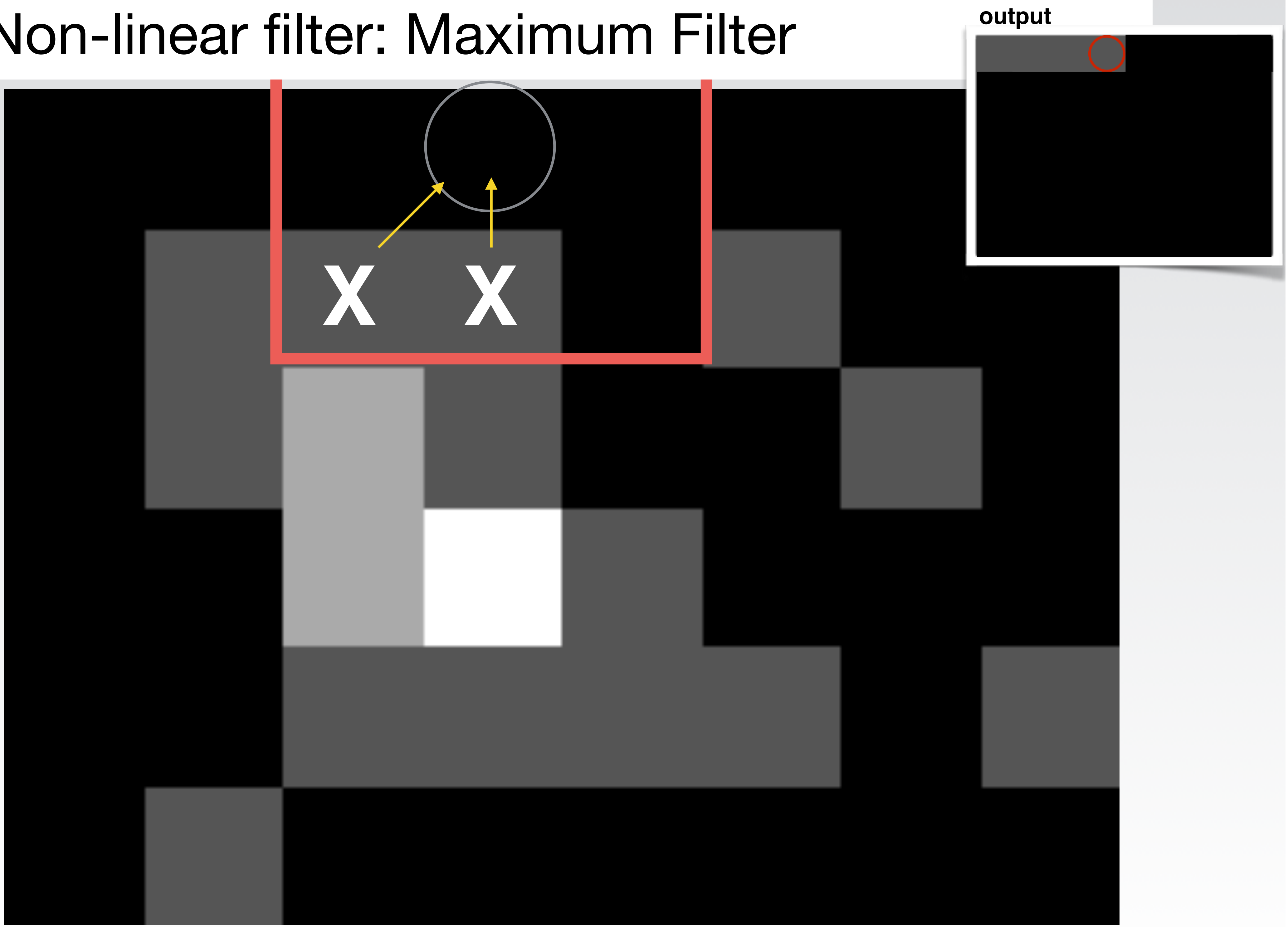




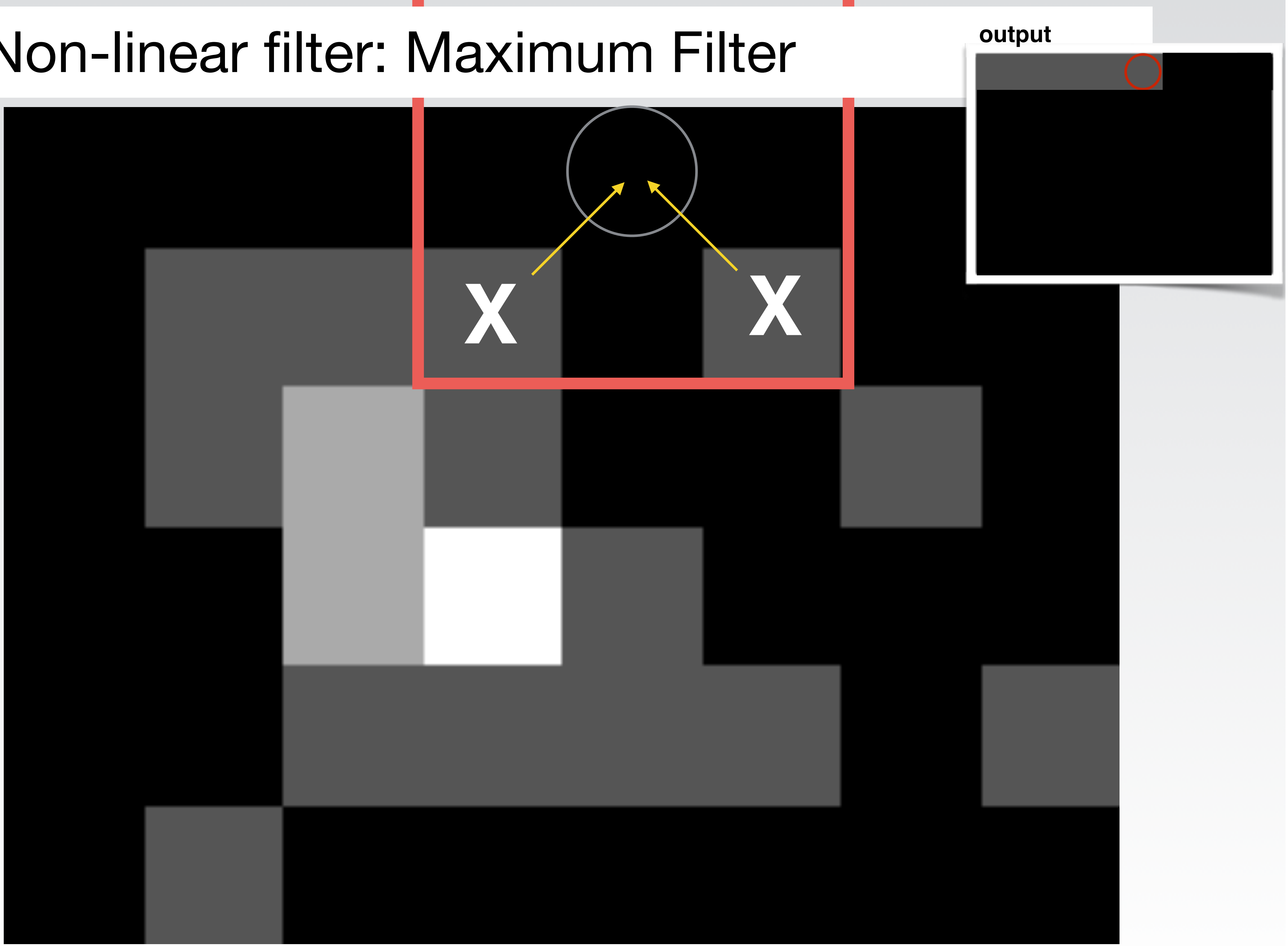
Non-linear filter: Maximum Filter



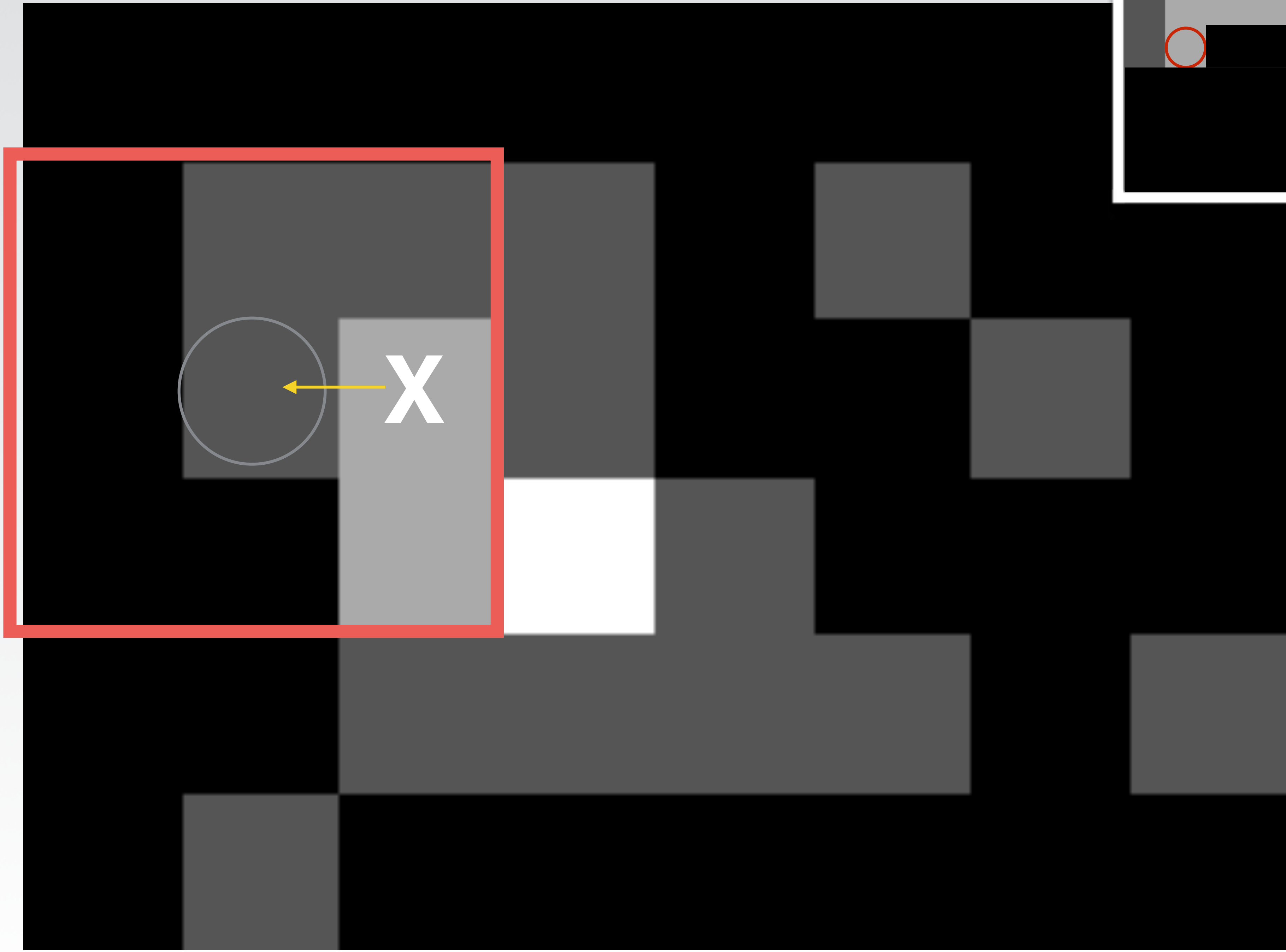
Non-linear filter: Maximum Filter

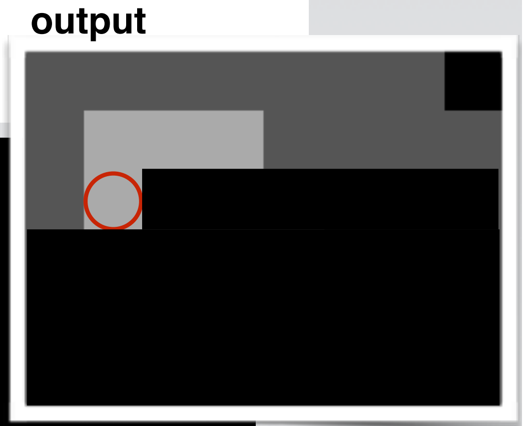


Non-linear filter: Maximum Filter

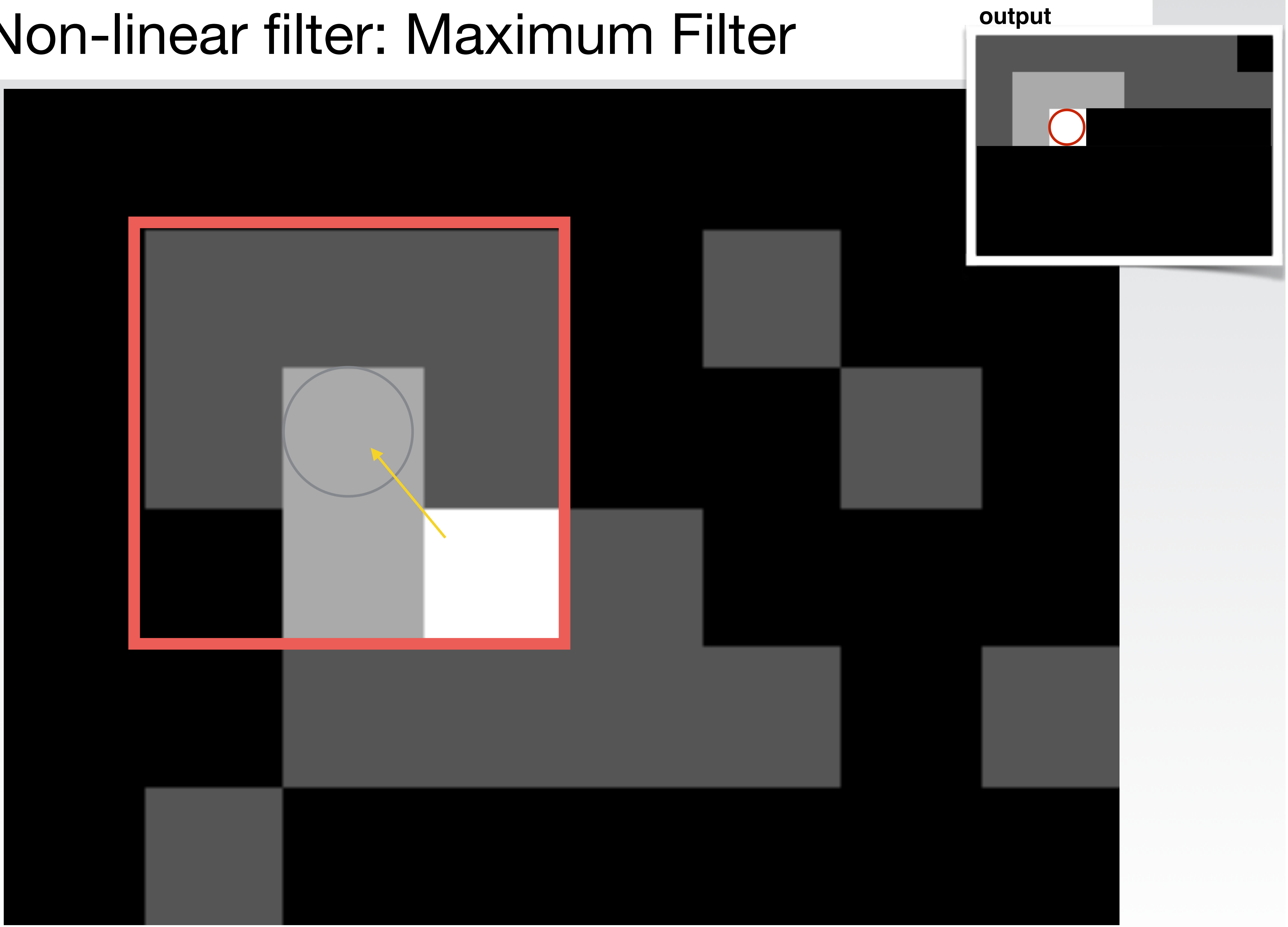


Non-linear filter: Maximum Filter

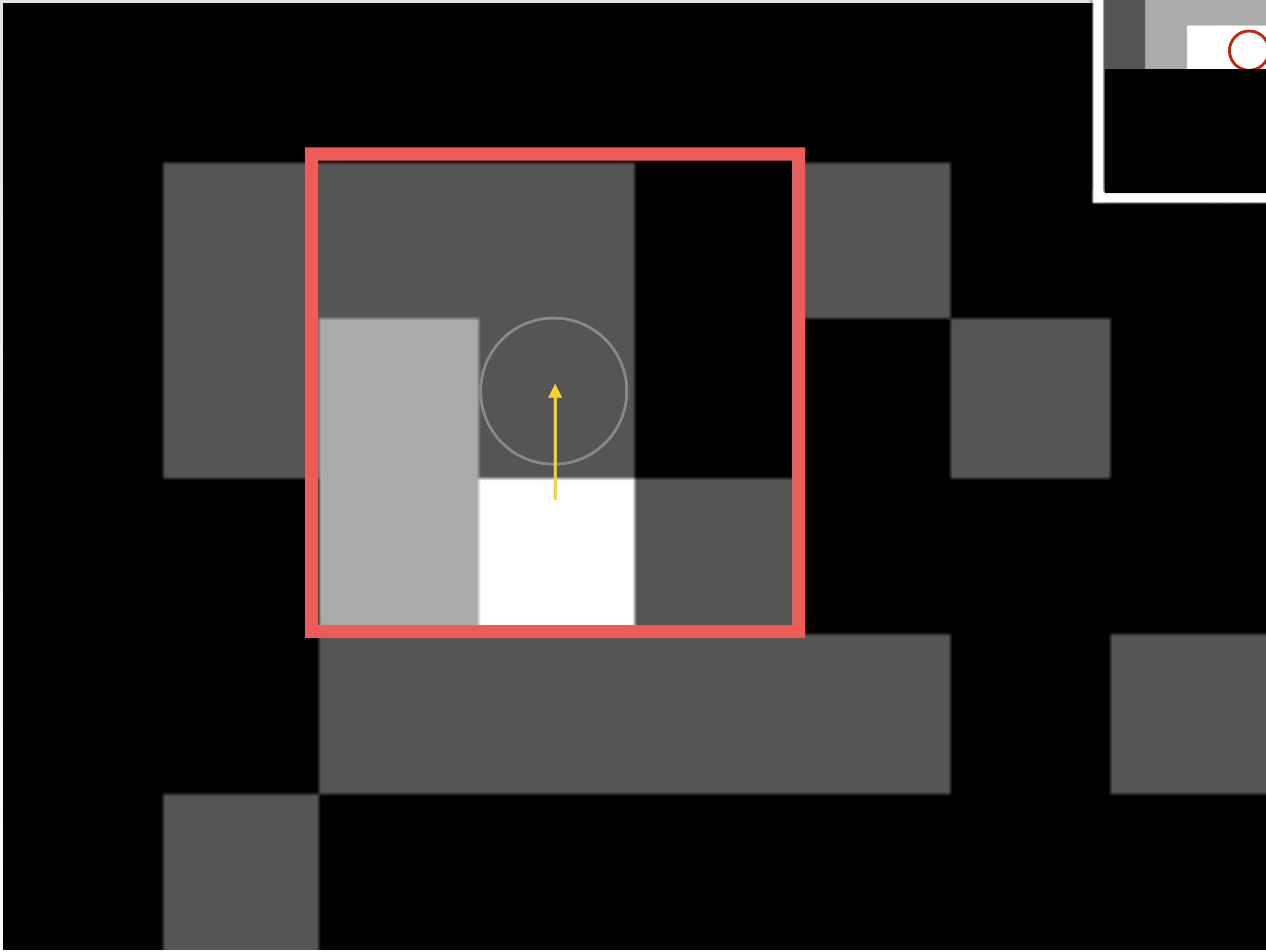


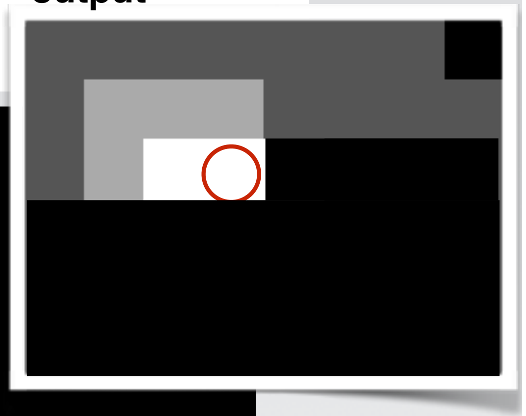


Non-linear filter: Maximum Filter

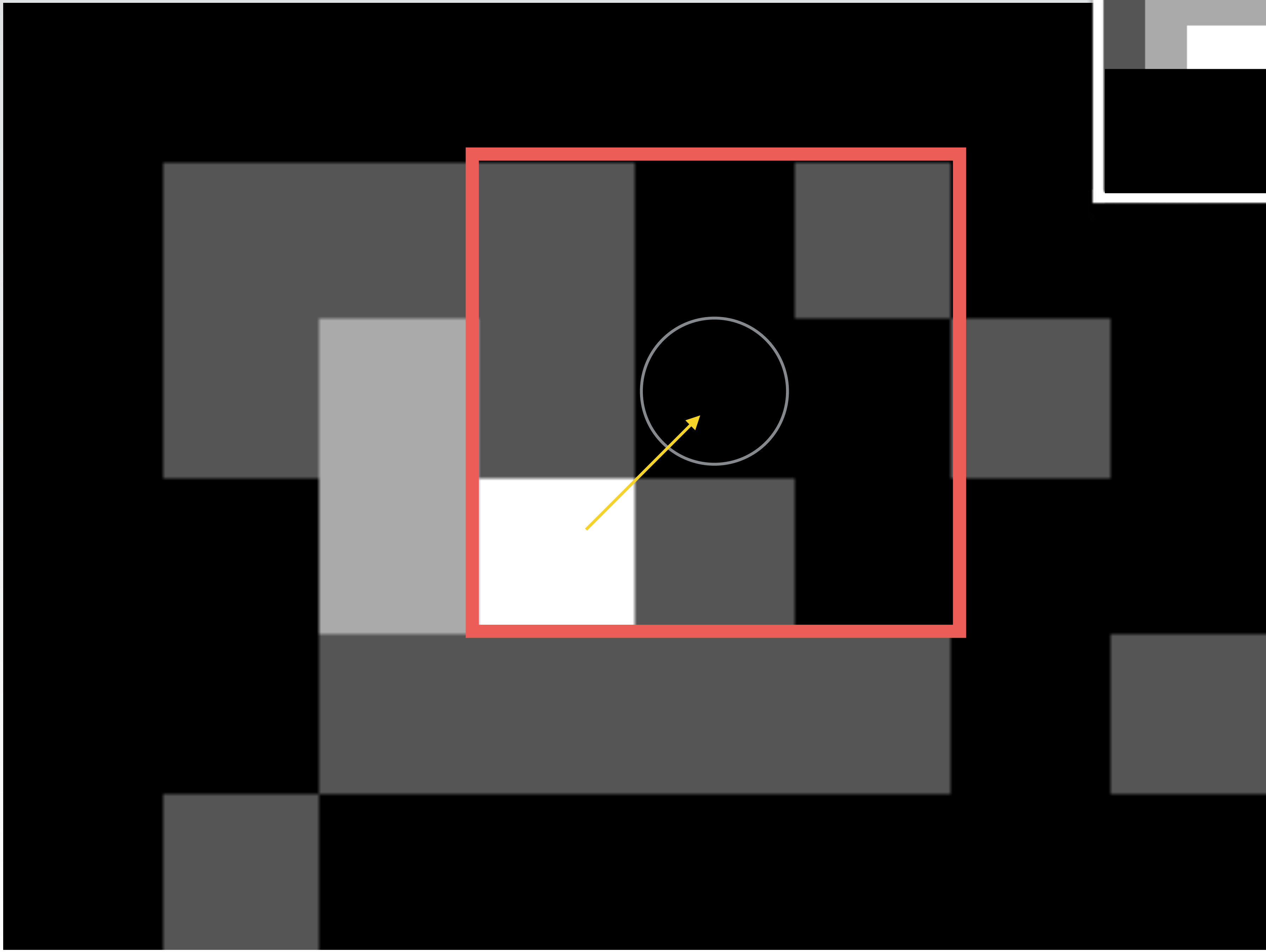


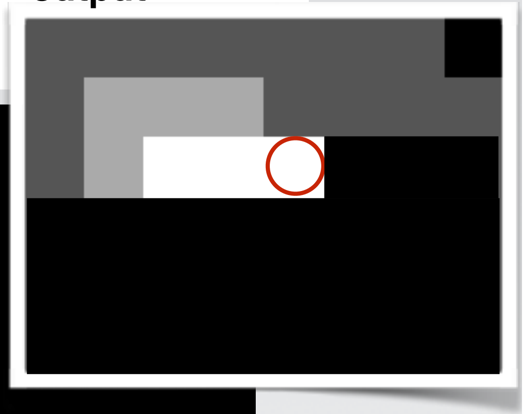
Non-linear filter: Maximum Filter



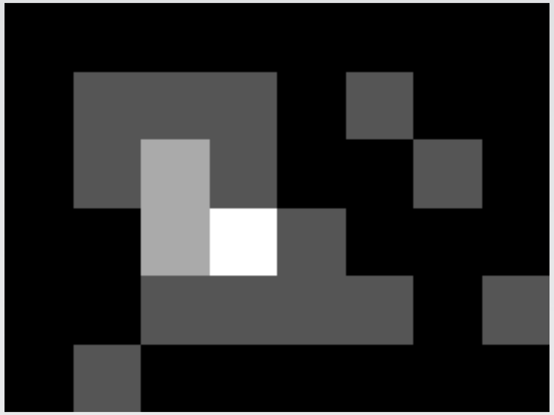


Non-linear filter: Maximum Filter

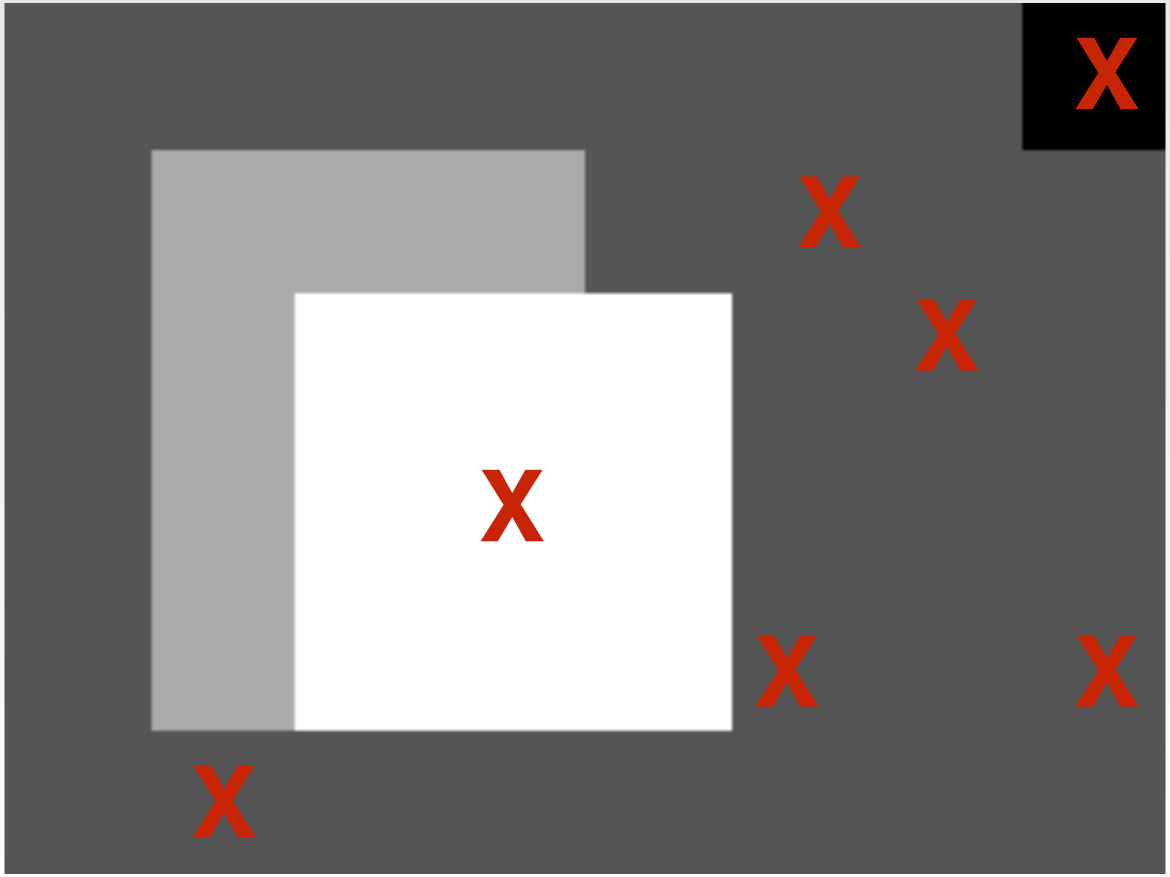




Result:



Where the pixel is the same in the input as the maximum filtered image you find your local maxima



Source:

Convolution - edge filters

1st derivative filter

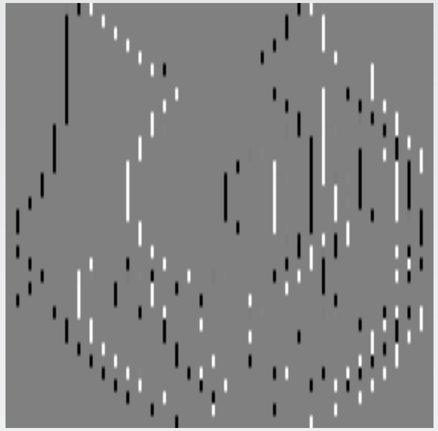
kernel

- 1
0
1

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

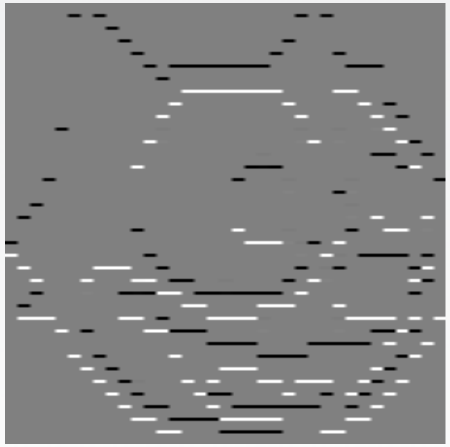
input

output (convolved)



other direction

- 1	0	1



Extracts edges in a particular direction

black is negative
white is positive

Source: http://www.uff.br/cdme/matrix/matrix-html/matrix_boolean/matrix_boolean_en.html

Convolution - edge filters

1st derivative filter 2d filter

kernel

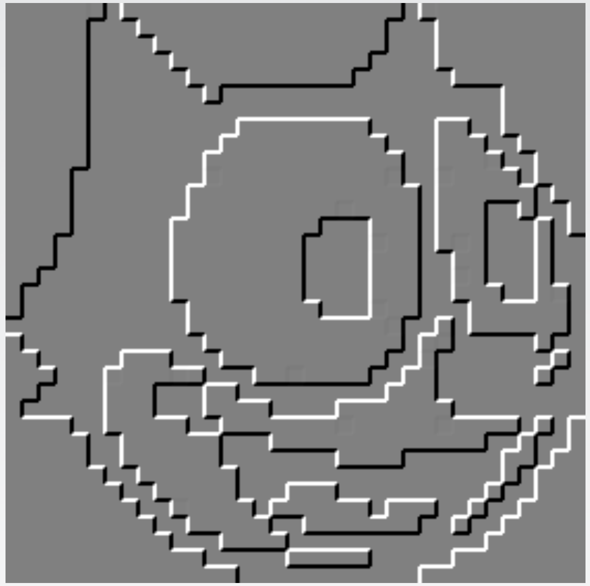
$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

2D filter

*

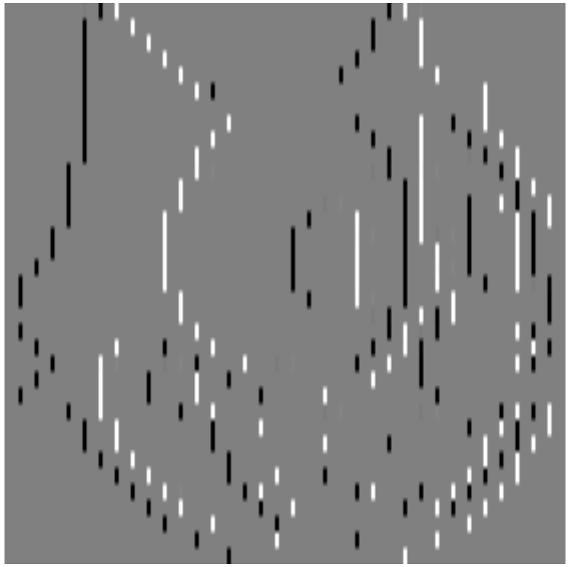
output (convolved)



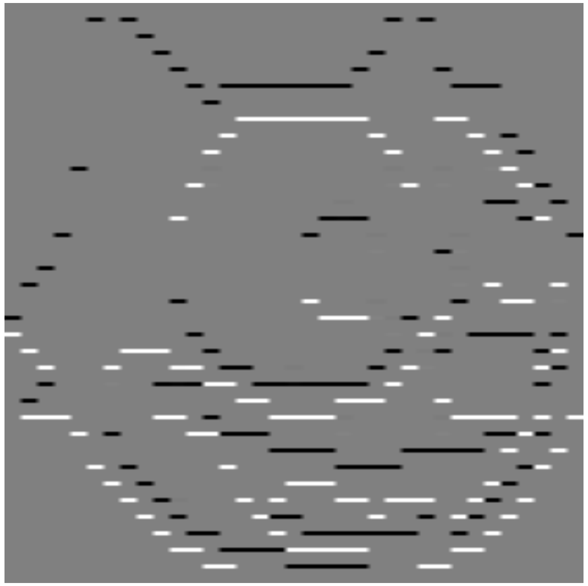
black is negative
white is positive

Source: http://www.uff.br/cdme/matrix/matrix-html/matrix_boolean/matrix_boolean_en.html

Convolution - Gradient magnitude give edges

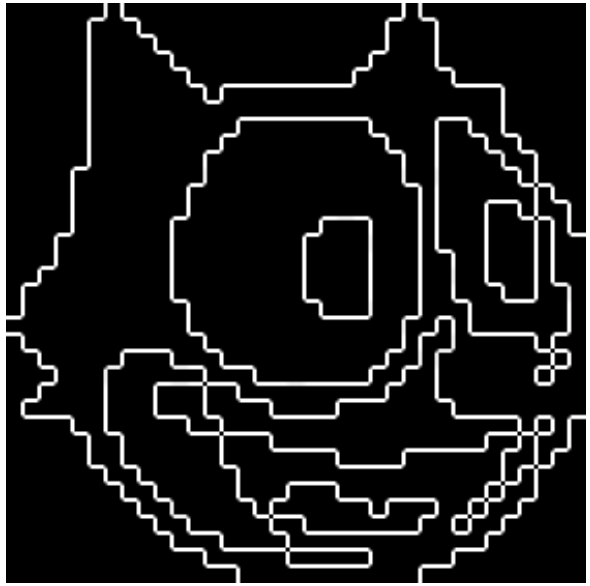


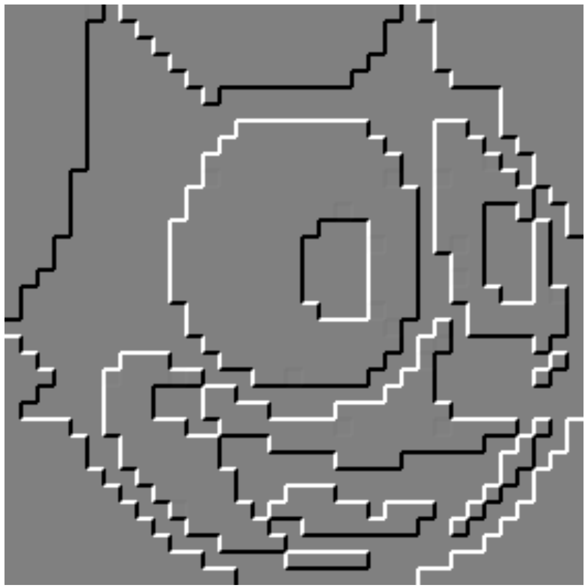
+



=

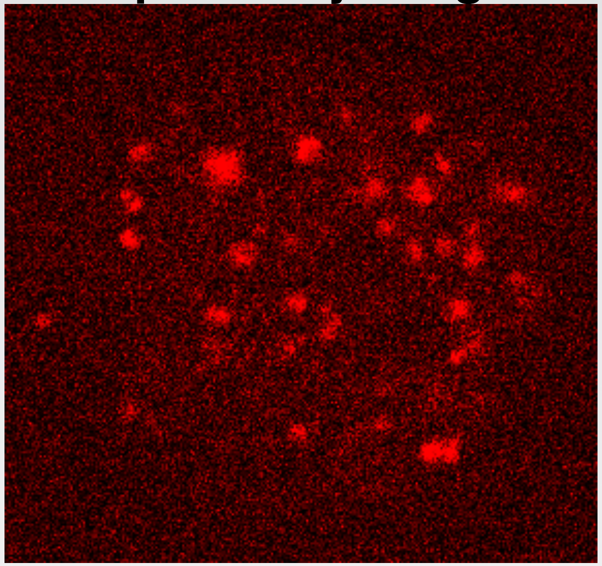
image**2





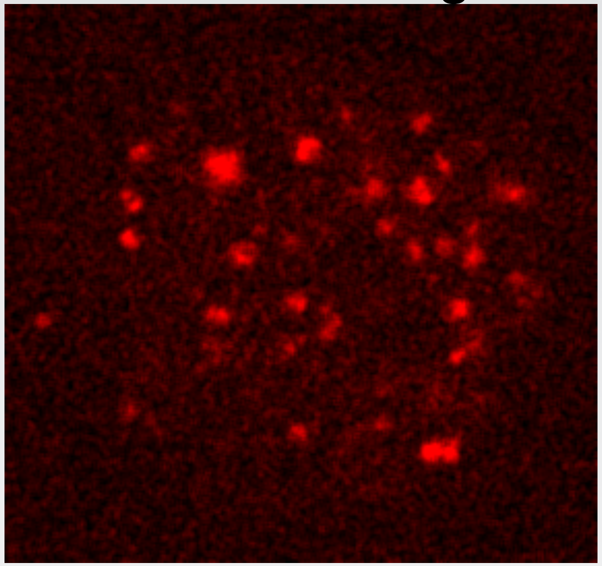
Mean Filter

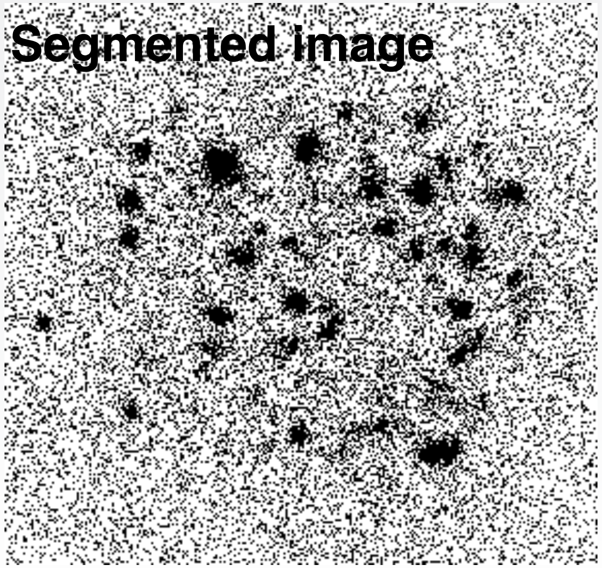
input noisy image

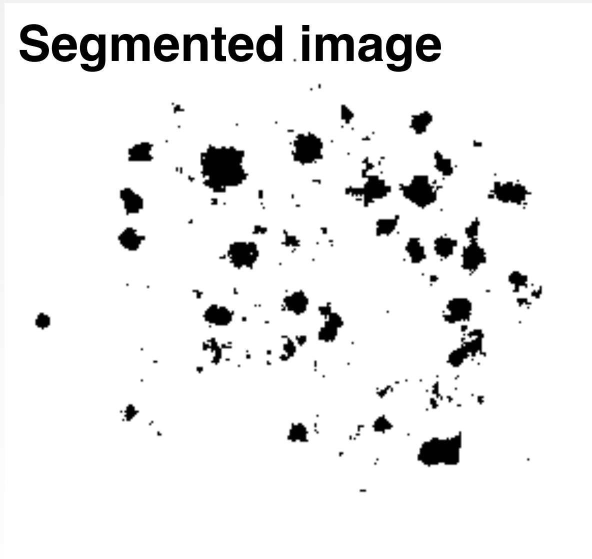


$$* \quad \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \quad =$$

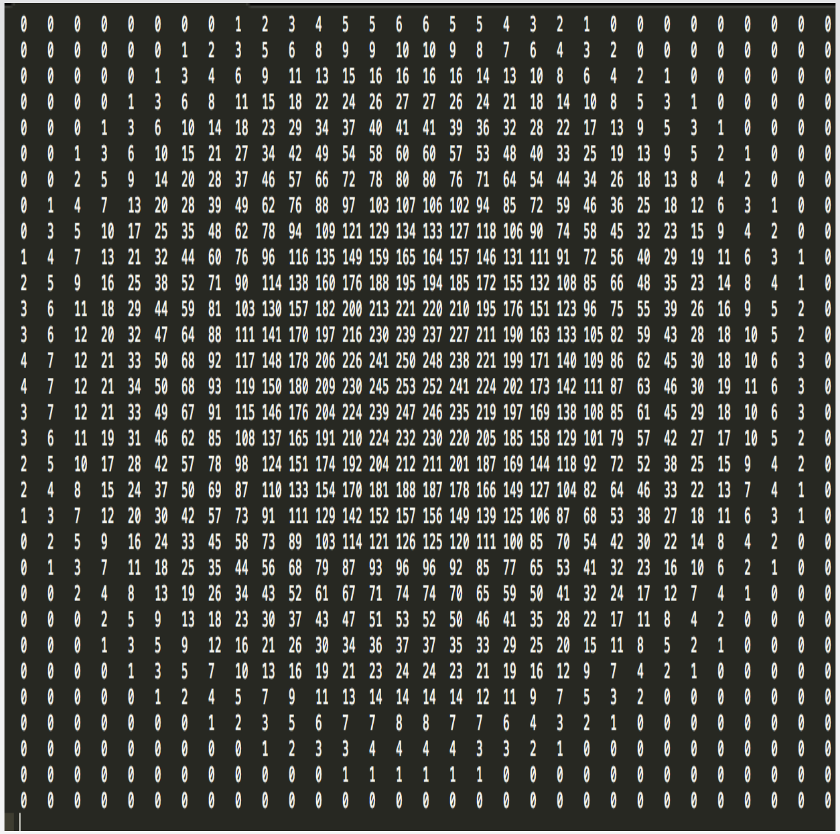
smoothed image



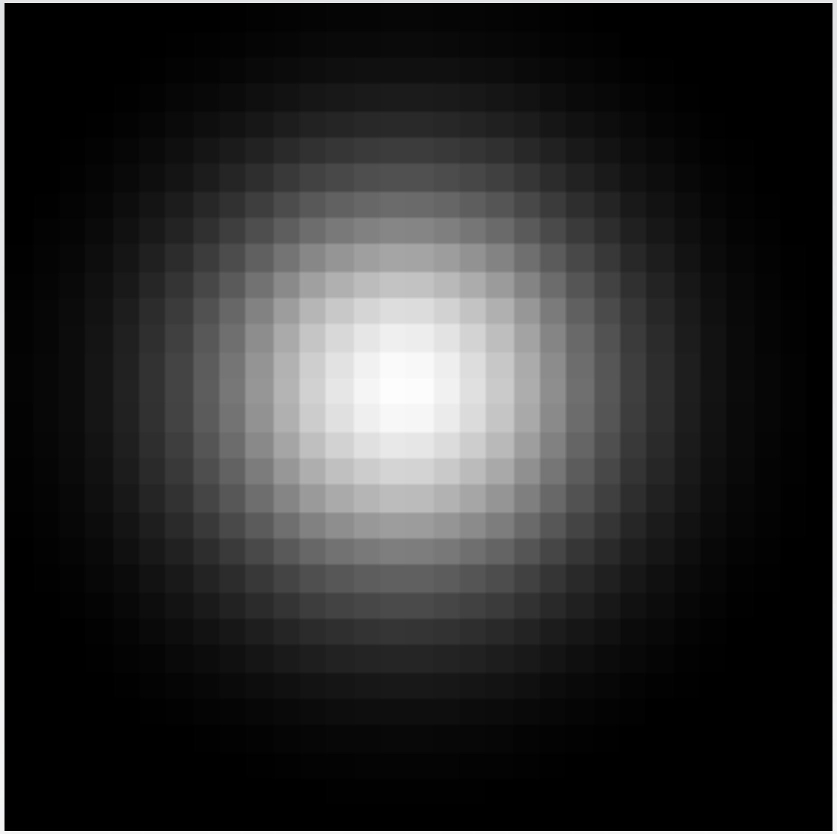




Gaussian filter.



0	0	0	0	0	0	0	0	1	2	3	4	5	5	6	6	5	5	4	3	2	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	2	3	5	6	8	9	9	10	10	9	8	7	6	4	3	2	0	0	0	0	0	0	0	0
0	0	0	0	0	1	3	4	6	9	11	13	15	16	16	16	16	14	13	10	8	6	4	2	1	0	0	0	0	0	0
0	0	0	0	1	3	6	8	11	15	18	22	24	26	27	27	26	24	21	18	14	10	8	5	3	1	0	0	0	0	0
0	0	0	1	3	6	10	14	18	23	29	34	37	40	41	41	39	36	32	28	22	17	13	9	5	3	1	0	0	0	0
0	0	1	3	6	10	15	21	27	34	42	49	54	58	60	60	57	53	48	40	33	25	19	13	9	5	2	1	0	0	0
0	0	2	5	9	14	20	28	37	46	57	66	72	78	80	80	76	71	64	54	44	34	26	18	13	8	4	2	0	0	0
0	1	4	7	13	20	28	39	49	62	76	88	97	103	107	106	102	94	85	72	59	46	36	25	18	12	6	3	1	0	0
0	3	5	10	17	25	35	48	62	78	94	109	121	129	134	133	127	118	106	90	74	58	45	32	23	15	9	4	2	0	0
1	4	7	13	21	32	44	60	76	96	116	135	149	159	165	164	157	146	131	111	91	72	56	40	29	19	11	6	3	1	0
2	5	9	16	25	38	52	71	90	114	138	160	176	188	195	194	185	172	155	132	108	85	66	48	35	23	14	8	4	1	0
3	6	11	18	29	44	59	81	103	130	157	182	200	213	221	220	210	195	176	151	123	96	75	55	39	26	16	9	5	2	0
3	6	12	20	32	47	64	88	111	141	170	197	216	230	239	237	227	211	190	163	133	105	82	59	43	28	18	10	5	2	0
4	7	12	21	33	50	68	92	117	148	178	206	226	241	250	248	238	221	199	171	140	109	86	62	45	30	18	10	6	3	0
4	7	12	21	34	50	68	93	119	150	180	209	230	245	253	252	241	224	202	173	142	111	87	63	46	30	19	11	6	3	0
3	7	12	21	33	49	67	91	115	146	176	204	224	239	247	246	235	219	197	169	138	108	85	61	45	29	18	10	6	3	0
3	6	11	19	31	46	62	85	108	137	165	191	210	224	232	230	220	205	185	158	129	101	79	57	42	27	17	10	5	2	0
2	5	10	17	28	42	57	78	98	124	151	174	192	204	212	211	201	187	169	144	118	92	72	52	38	25	15	9	4	2	0
2	4	8	15	24	37	50	69	87	110	133	154	170	181	188	187	178	166	149	127	104	82	64	46	33	22	13	7	4	1	0
1	3	7	12	20	30	42	57	73	91	111	129	142	152	157	156	149	139	125	106	87	68	53	38	27	18	11	6	3	1	0
0	2	5	9	16	24	33	45	58	73	89	103	114	121	126	125	120	111	100	85	70	54	42	30	22	14	8	4	2	0	0
0	1	3	7	11	18	25	35	44	56	68	79	87	93	96	96	92	85	77	65	53	41	32	23	16	10	6	2	1	0	0
0	0	2	4	8	13	19	26	34	43	52	61	67	71	74	74	70	65	59	50	41	32	24	17	12	7	4	1	0	0	0
0	0	0	2	5	9	13	18	23	30	37	43	47	51	53	52	50	46	41	35	28	22	17	11	8	4	2	0	0	0	0
0	0	0	1	3	5	9	12	16	21	26	30	34	36	37	37	35	33	29	25	20	15	11	8	5	2	1	0	0	0	0
0	0	0	0	1	3	5	7	10	13	16	19	21	23	24	24	23	21	19	16	12	9	7	4	2	1	0	0	0	0	0
0	0	0	0	0	1	2	4	5	7	9	11	13	14	14	14	14	12	11	9	7	5	3	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	2	3	5	6	7	7	8	8	7	7	6	4	3	2	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	2	3	3	4	4	4	4	3	3	2	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	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	0	0	0	0	0	0	0	0

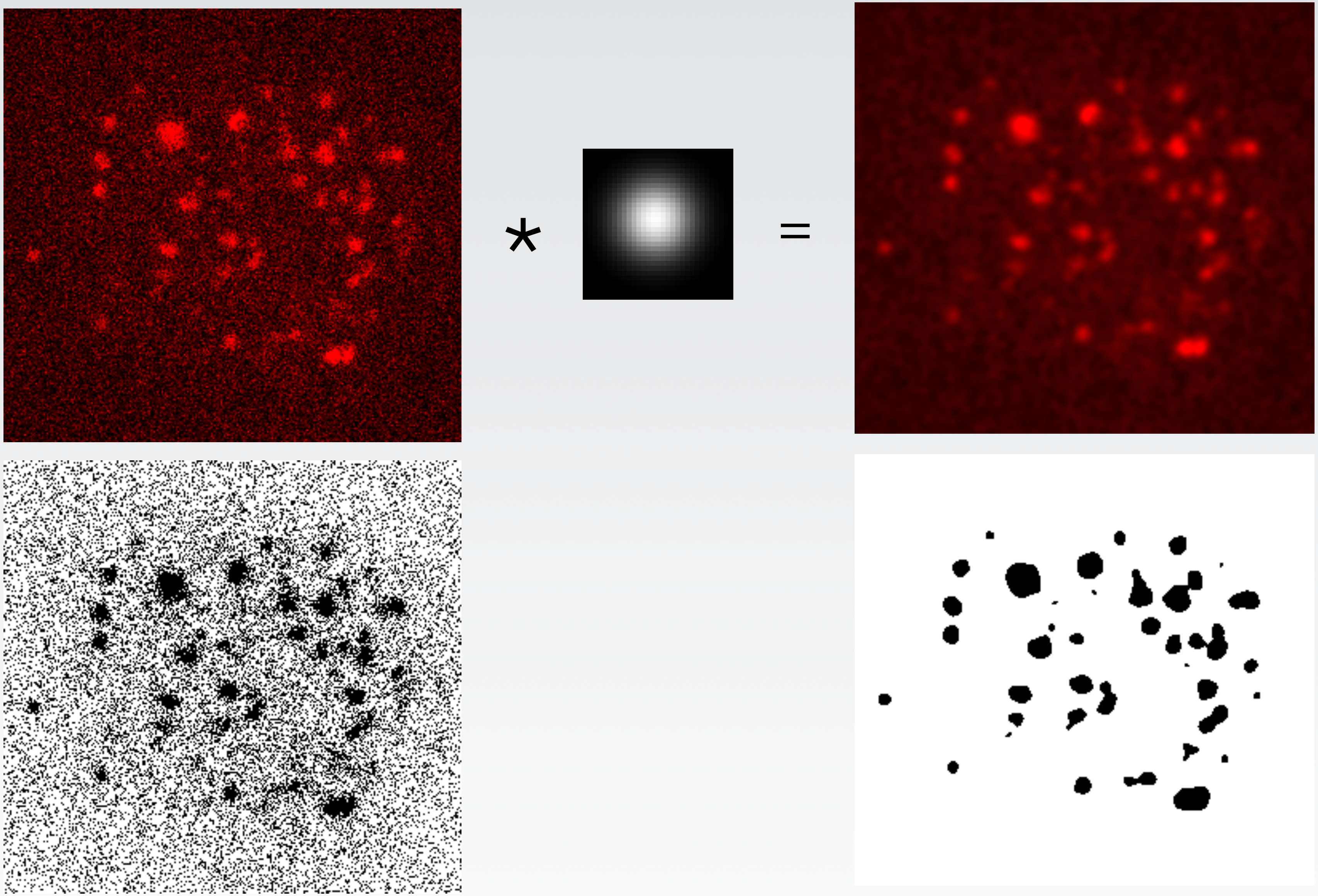


$$f(x,y) = A\exp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(y-y_o)^2}{2\sigma_y^2}\right)\right)$$

x, y (pixel coordinates)
sigma of kernel
x0,y0 center of Gaussian

We can use this to remove the image background.

Denoising through Gaussian deconvolution



Weights pixels nearer centre more

Convolution - Laplacian filter

2nd derivate filter

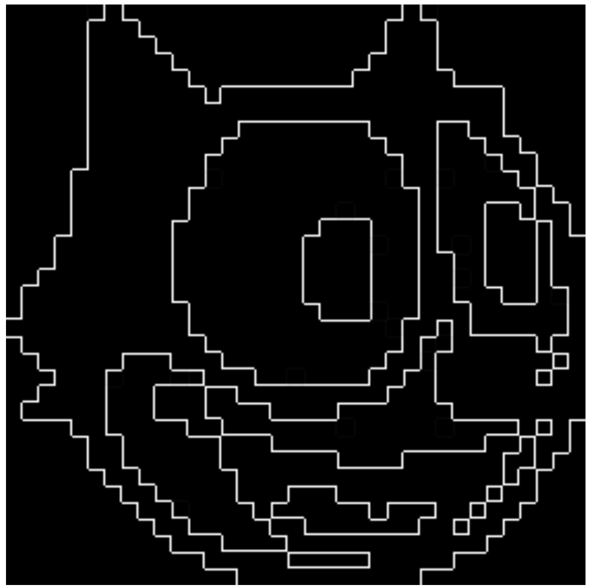
$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

1D filter: $\vec{D}_x^2 = [1 \quad -2 \quad 1]$,

2D filter: $\mathbf{D}_{xy}^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$.

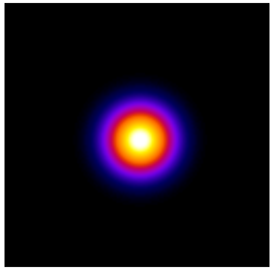
2D filter: $\mathbf{D}_{xy}^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$.

2-D laplacian filter detects change no matter in which direction. Wherever the rate of change is large.

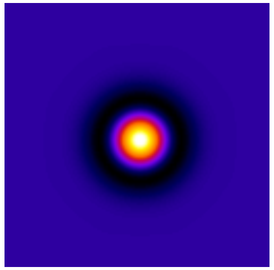


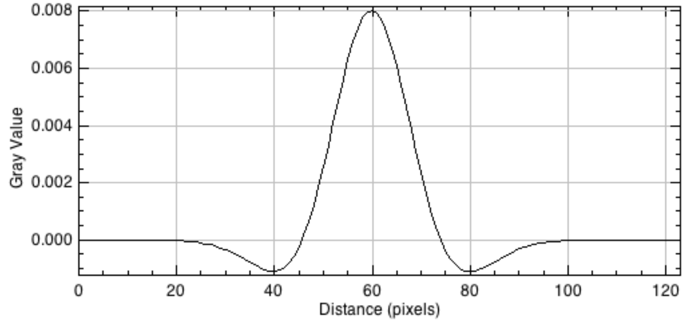
Gaussian of laplacian filters

In practice discrete filters are noisy.



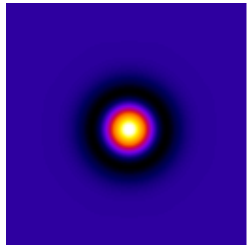
$$* \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} =$$



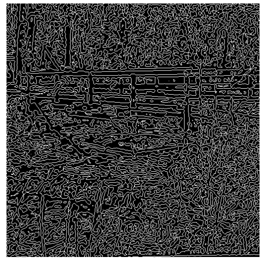


$$* \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} =$$

$*$



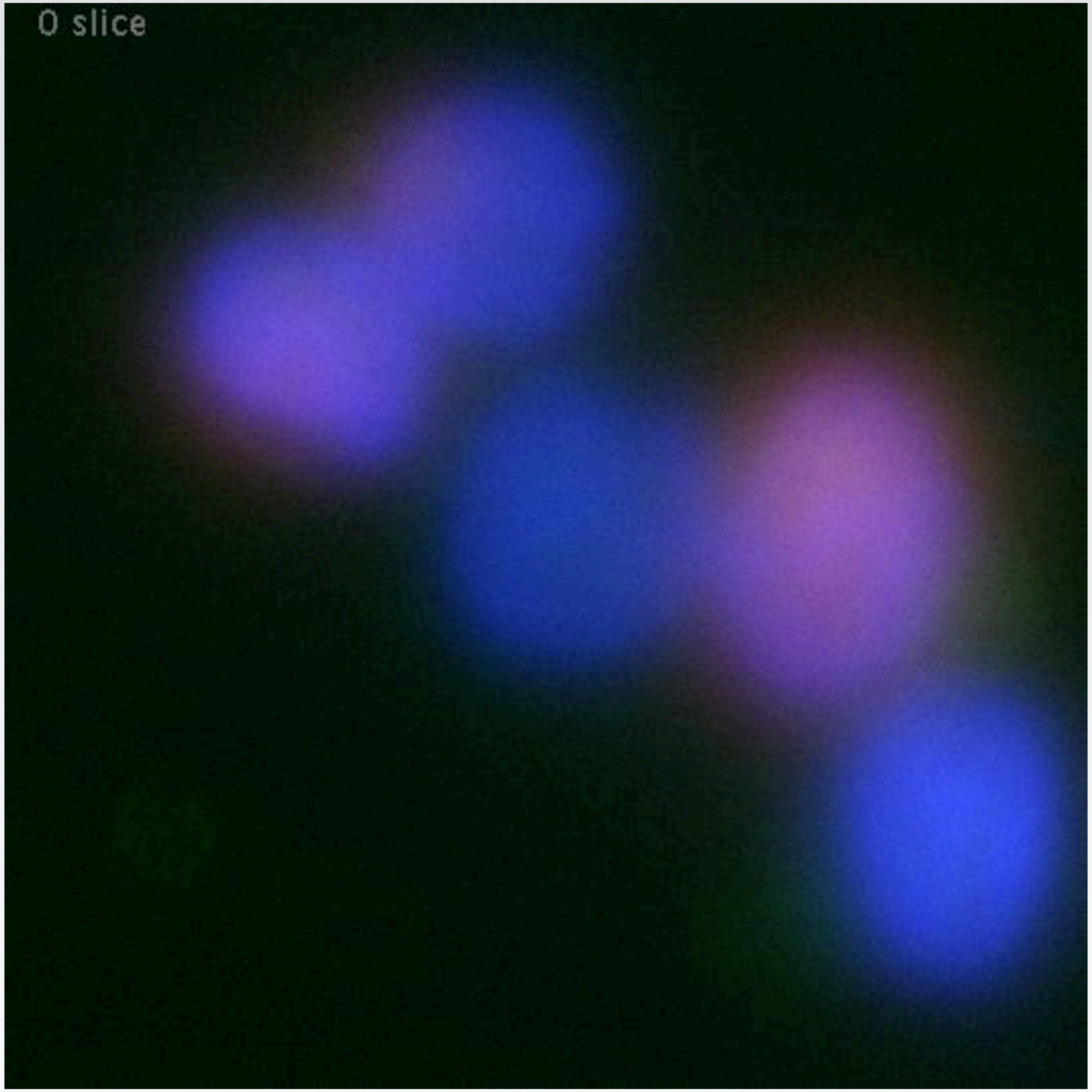
$=$



zero crossing

We can prevent the effects of this by convolving our filter (or our image first with a Gaussian filter).

Motivation for Analysis



Acquired on a Deltavision wide-field microscope.

Multiple z-slices.

The data is very rich, 3-D.

We want to find the colocalisation of the ATRX protein (Green) and the YH2Ax protein (Red)

Source: David Clynes

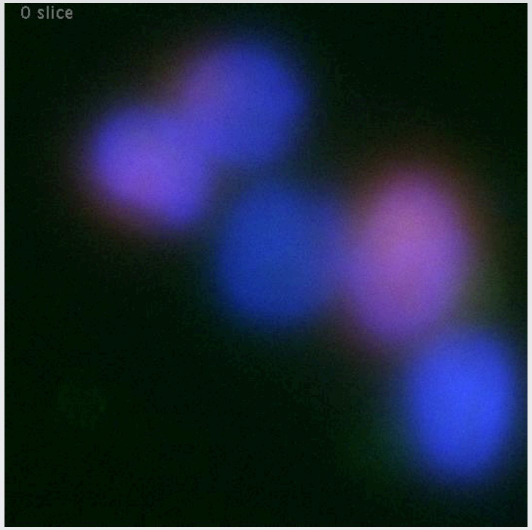
Motivation for Analysis

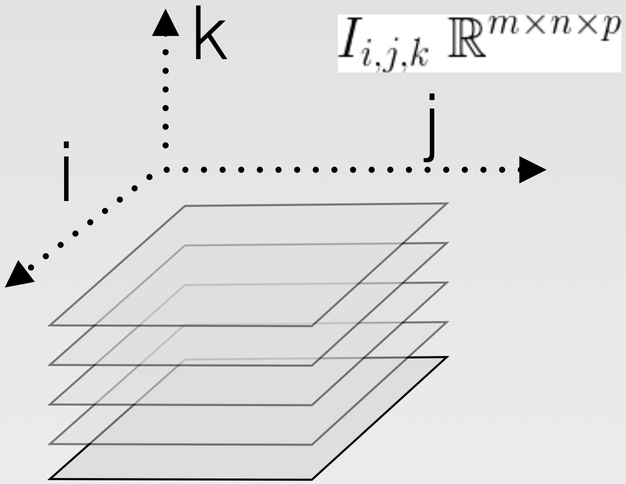
Early in your experimental pipeline you need to decide what the role of analysis is in your work.

- Is your research focus about developing novel analysis to solve a problem.
- Is your research about using analysis to investigate a biological question.

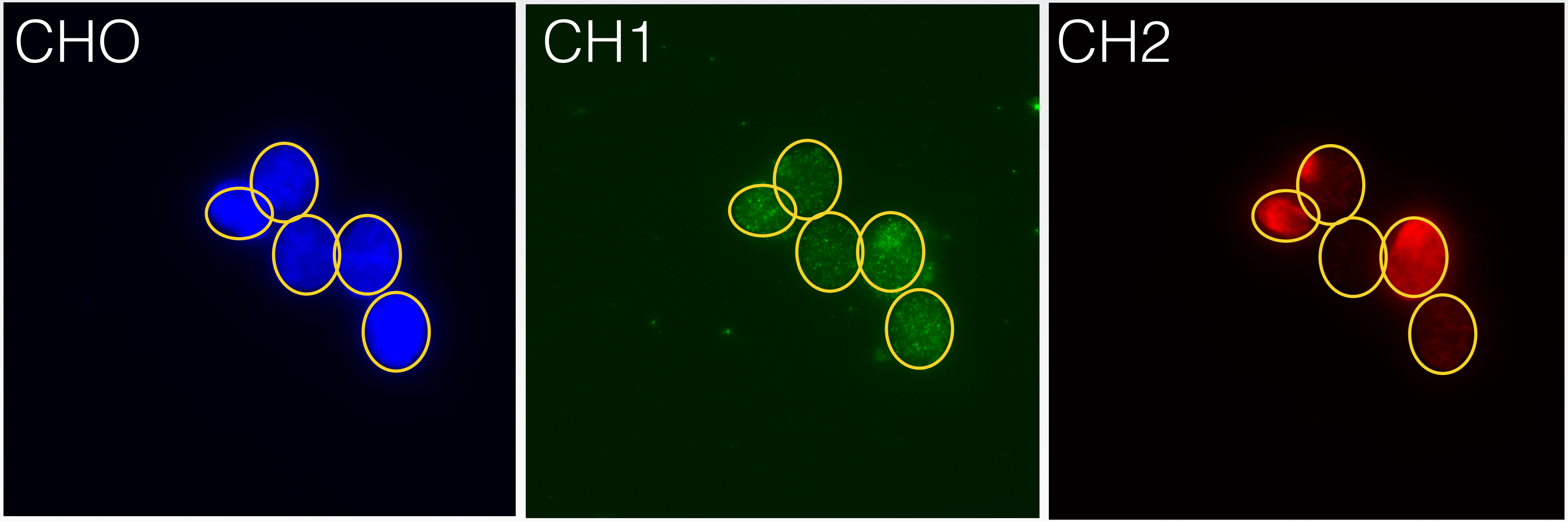
In this case we are just interested in creating a read-out for the biological experiment which is efficient and robust.

Typical imaging experiment



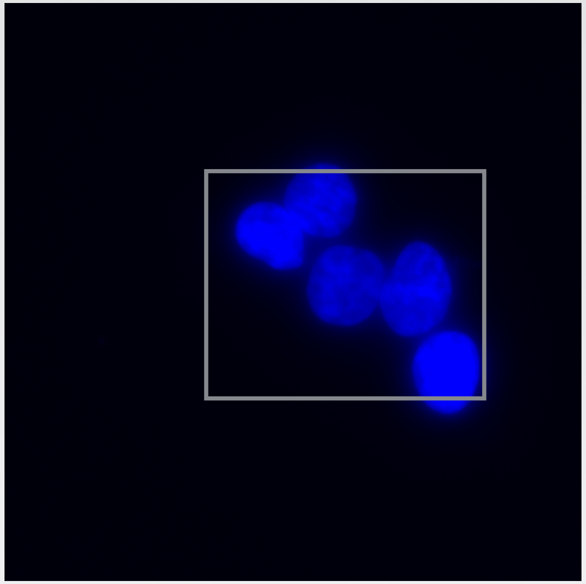


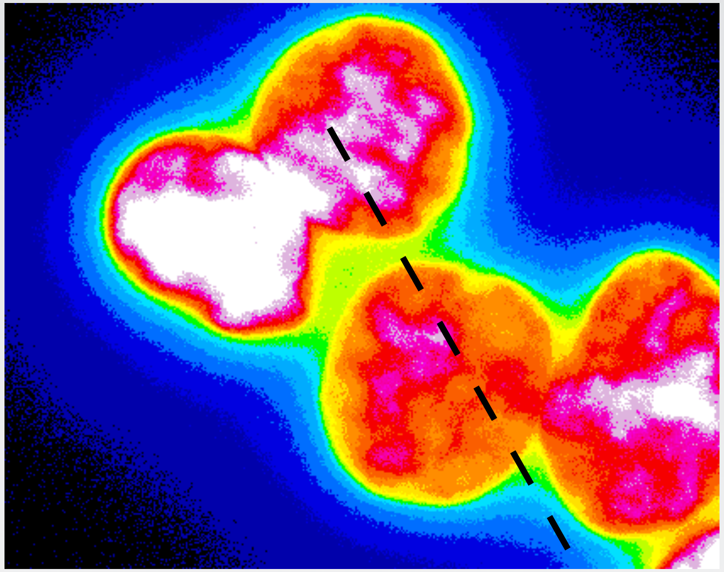
$$I_{i,j} = \max_{k=1}^{P}(I_{i,j,k})$$

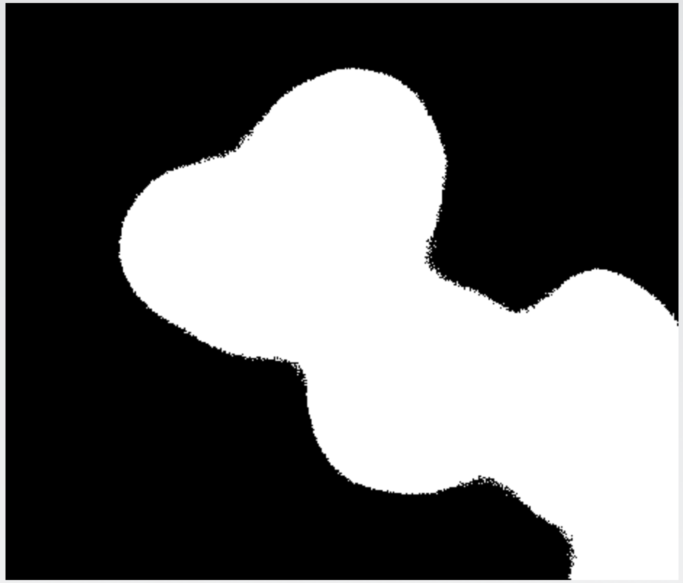


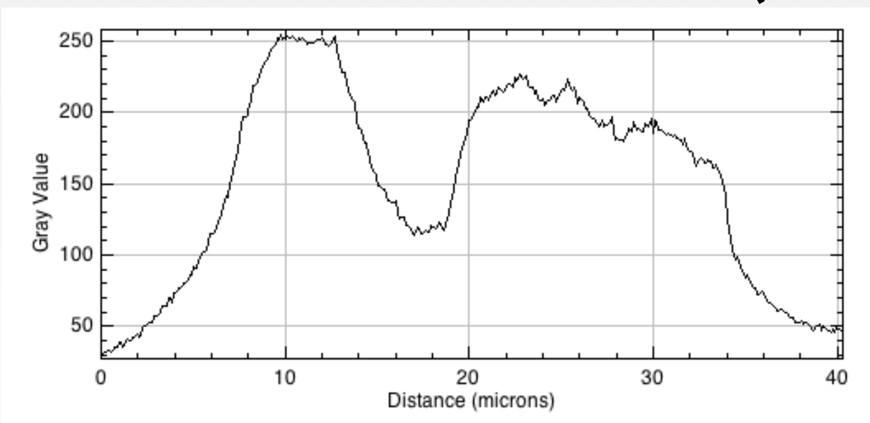
We want to isolate image regions into individual cells and then make measurements.

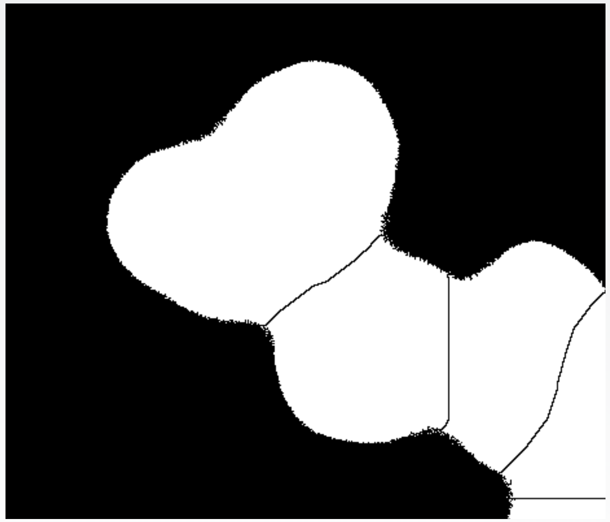
High background signal





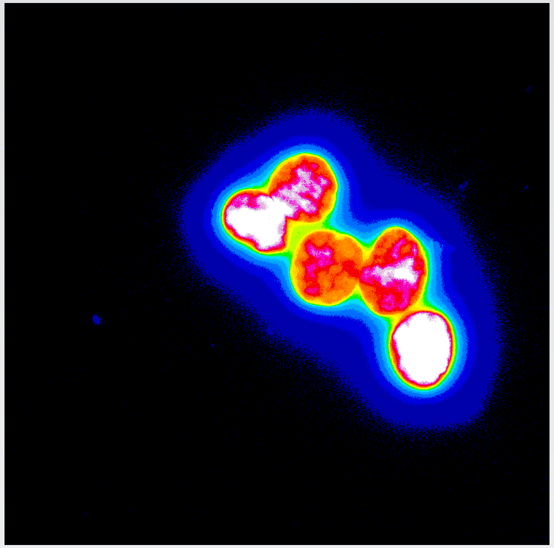




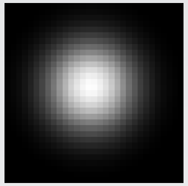


Cells will be hard to separate. We need to remove this smooth background signal. This background signal is low-frequency image signal.

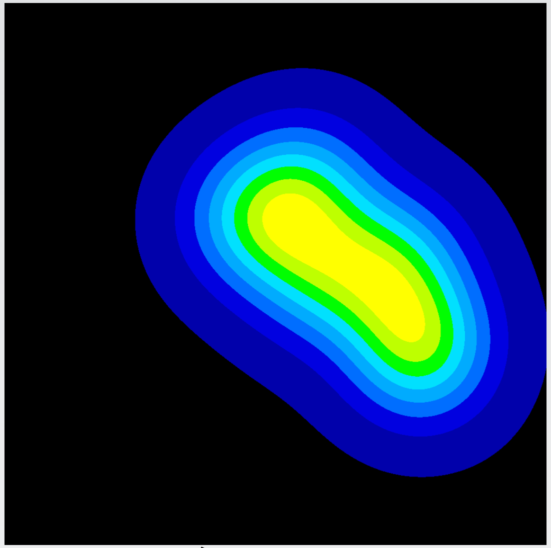
Background removal, using filtering



CONVOLVE



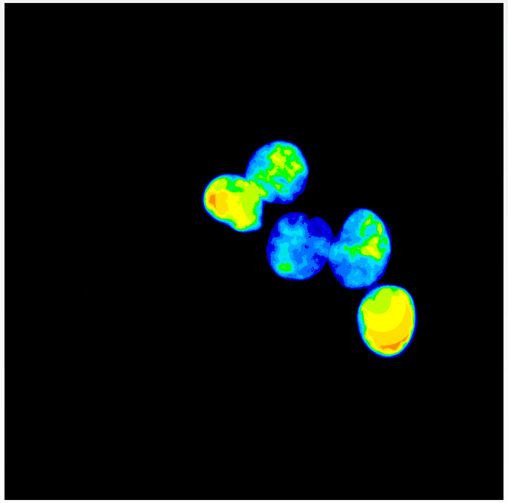
Gaussian heavy
sigma = 80



Low frequencies of image.

equivalent to a low pass filter

SUBTRACT



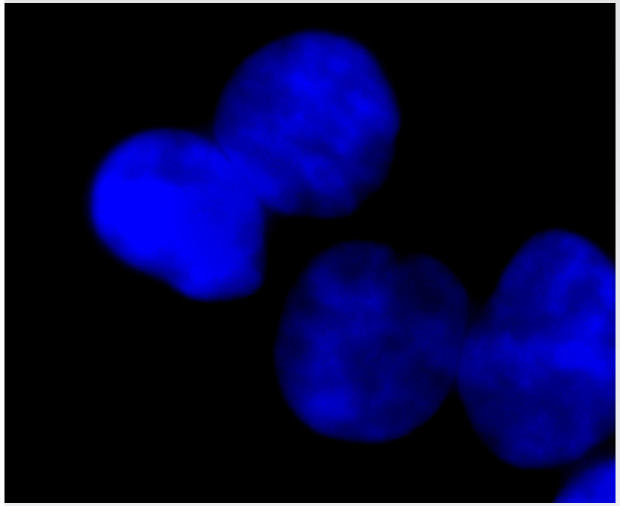
This is the resulting image.

The low frequencies have been removed leaving all the other frequencies.

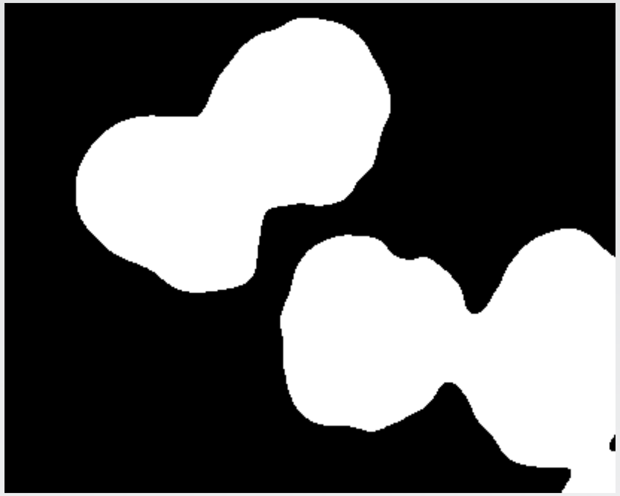
Equivalent to a high pass filter.

Source:

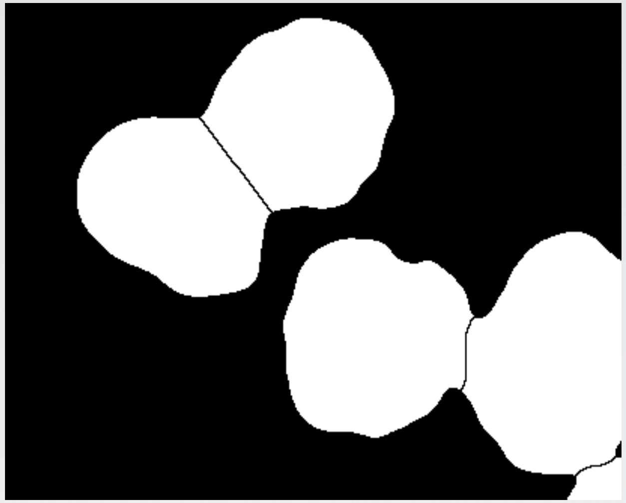
Segmentation much improved.



Filtered image



Segmented image

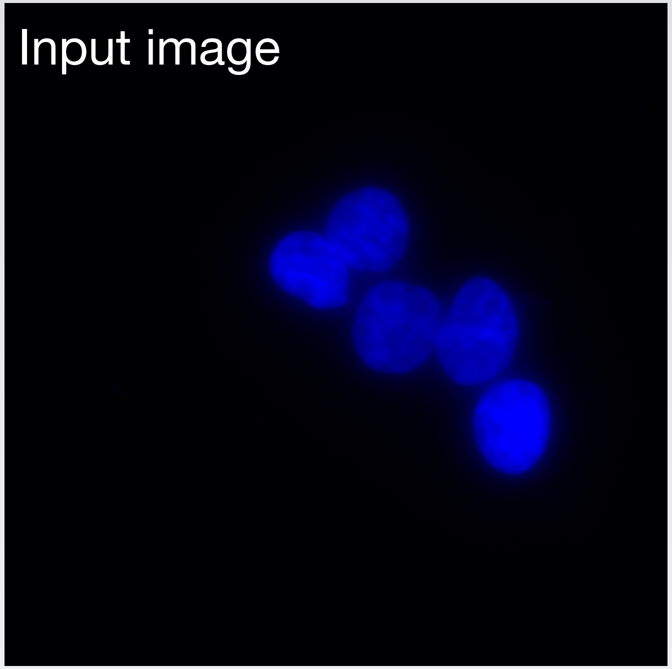


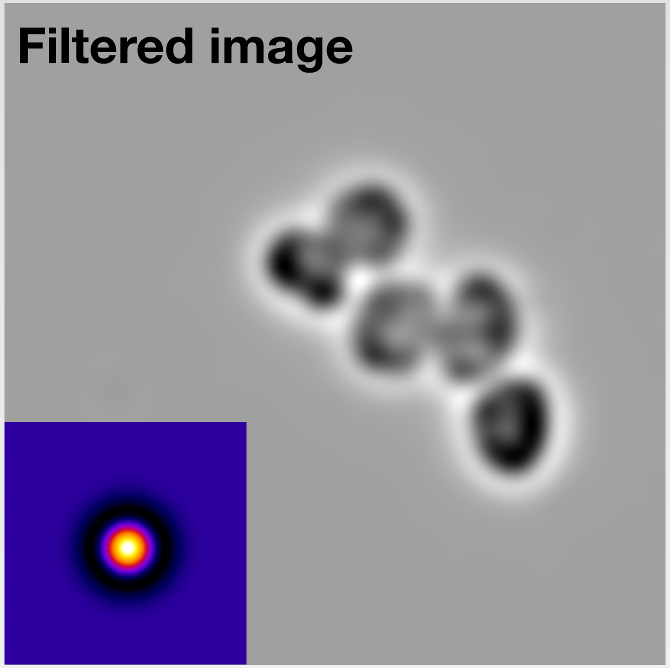
Watershedded image

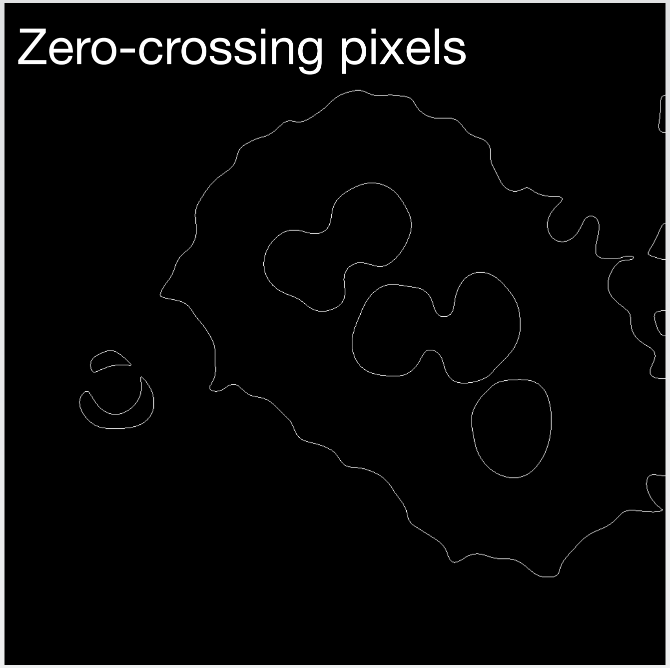
Segmented cells

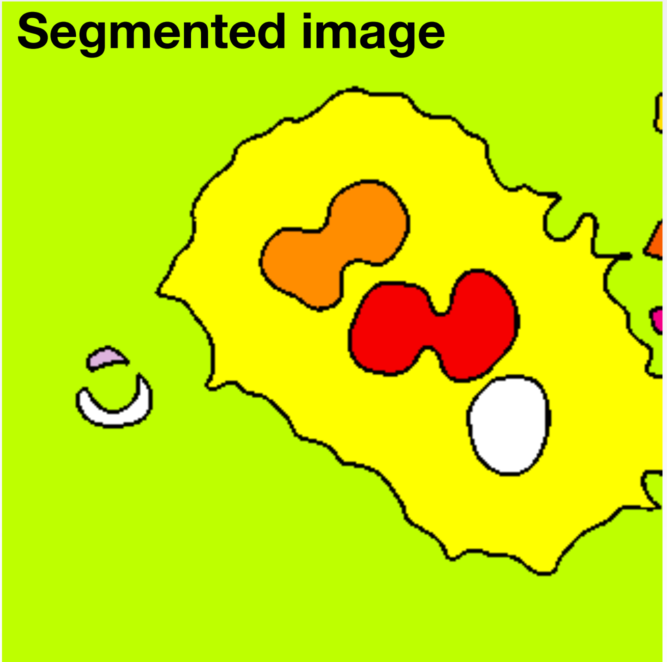
Source:

Smoothed laplacian filter







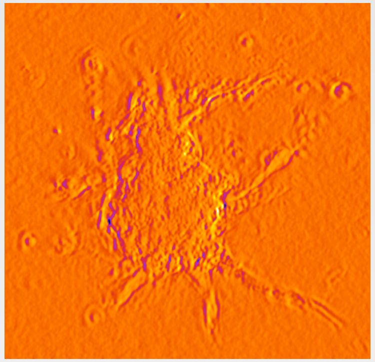


Source:

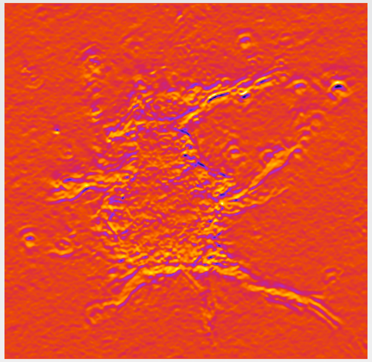
Feature calculation

If you filter and image with

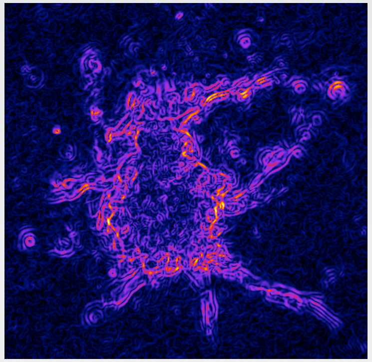
derivative-X



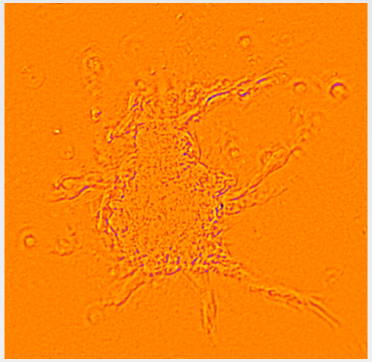
derivative-Y



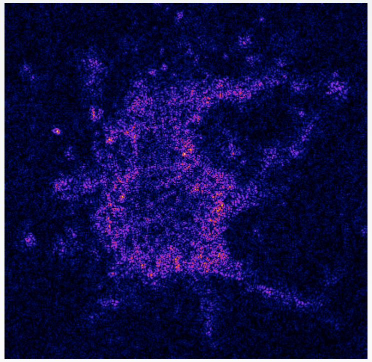
Edge mag.



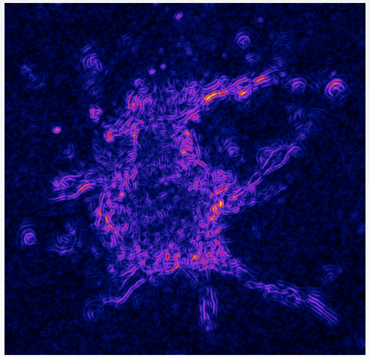
Laplacian



Hessian min

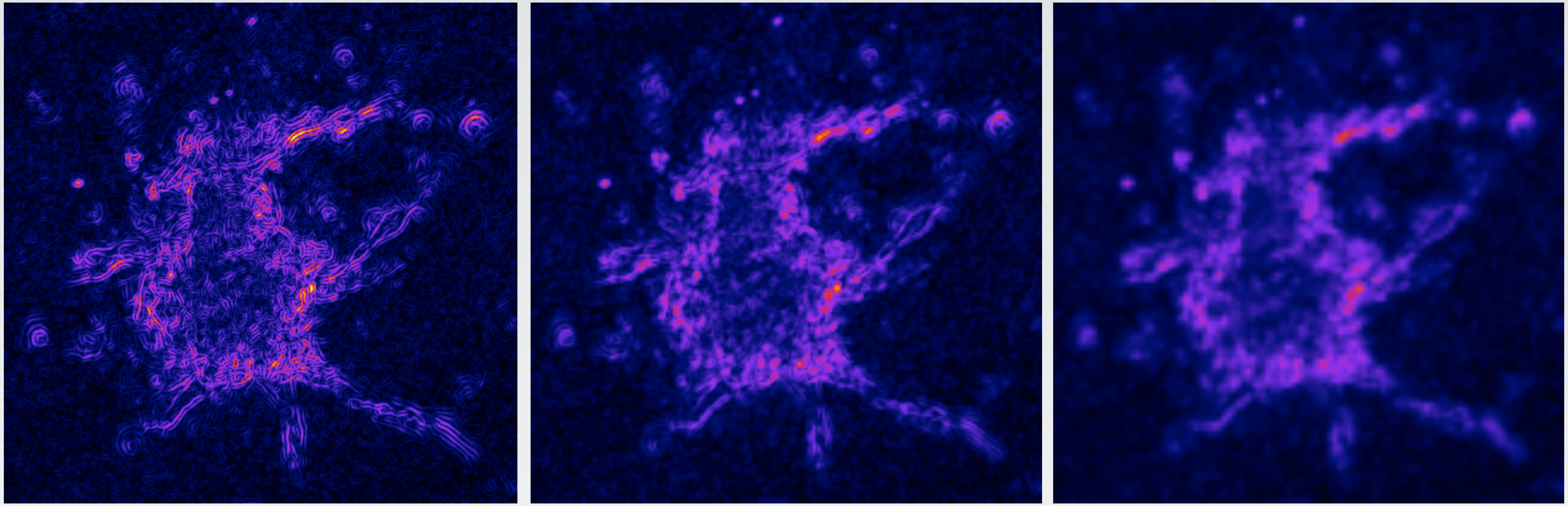


Hessian max



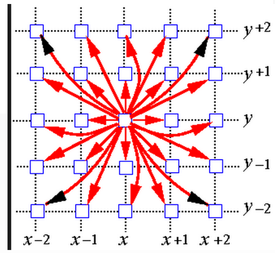
Source:FeatureJ

Feature Calculation different scales



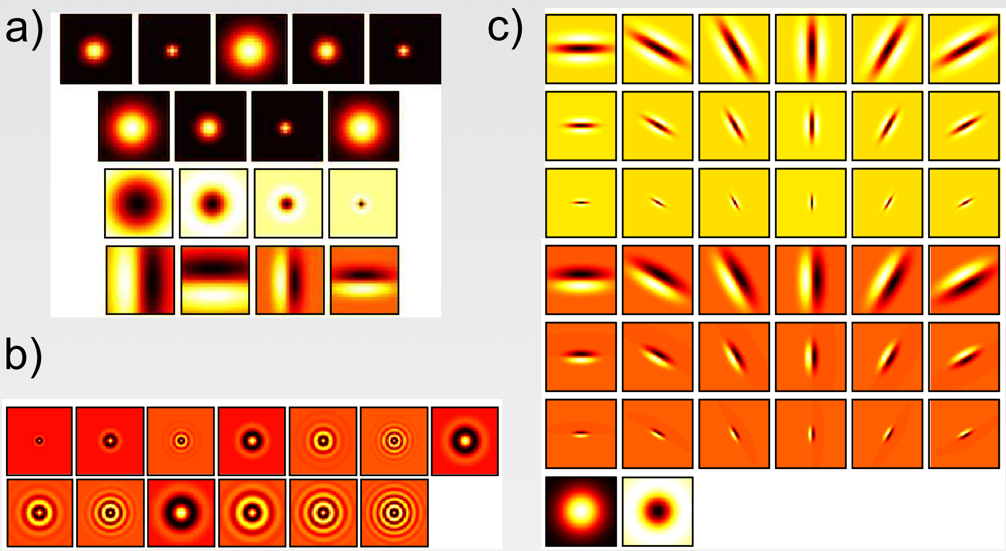
sigma = 1.0 **sigma = 4.0** **sigma = 8.0**

By blurring the image and applying the different filters you get a sense of the greater neighbourhood. Many scales can be used in combination.



Source:

Filter banks



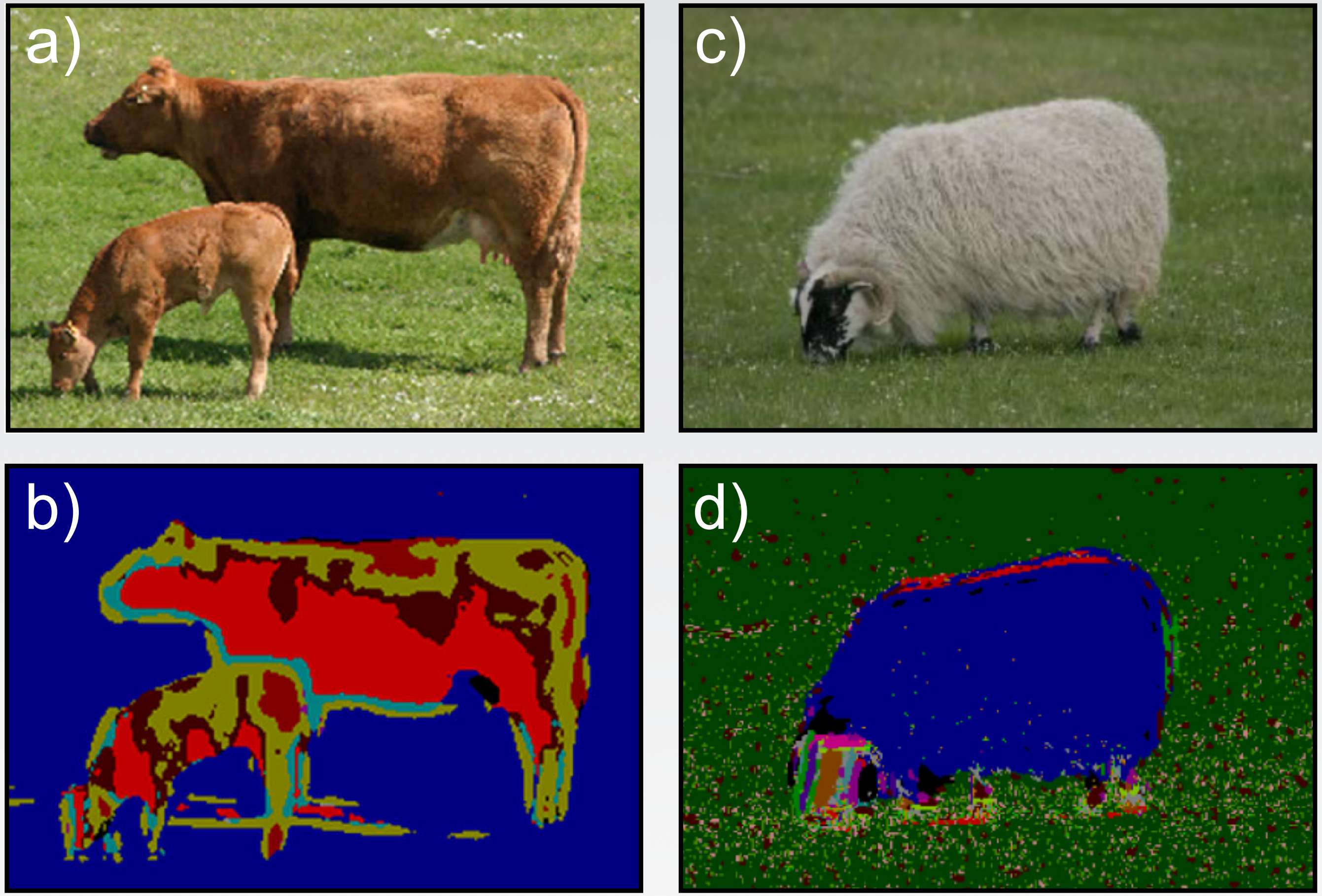
a) A combination of Gaussians, derivatives of Gaussians, Laplacians of Gaussians

b) Some rotationally invariant filters.

c) The maximum response MR8 filter bank

Source: Computer Vision, Simon Prince

Filters banks used for semantic segmentation



a) A combination of Gaussians, derivatives of Gaussians, Laplacians of Gaussians

b) Some rotationally invariant filters.

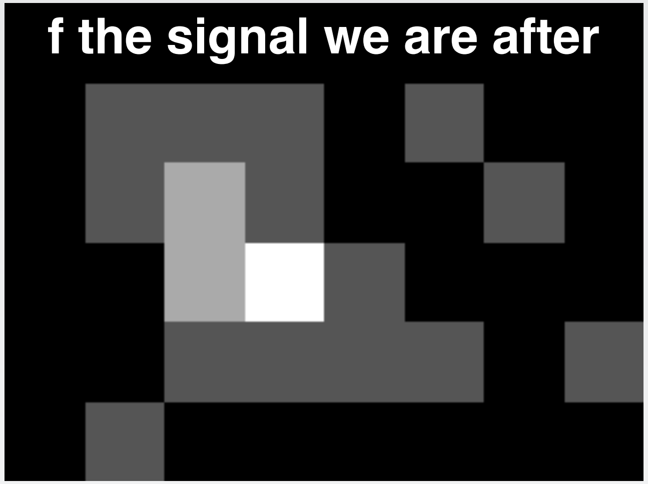
c) The maximum response MR8 filter bank

Source: Computer Vision, Simon Prince

Deconvolution

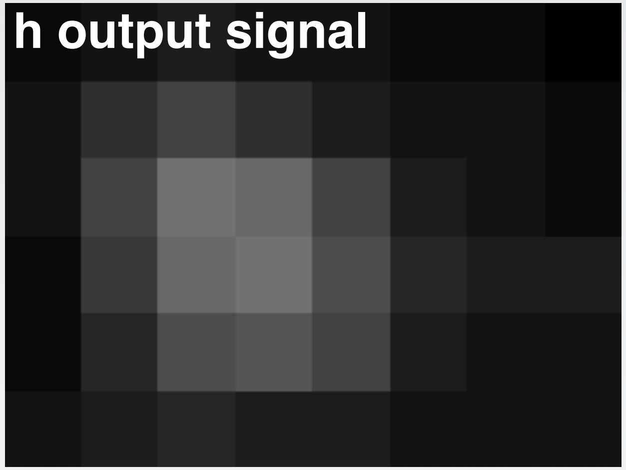
$$f * g = h$$

convolution



$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

kernel: 3x3 mean filter

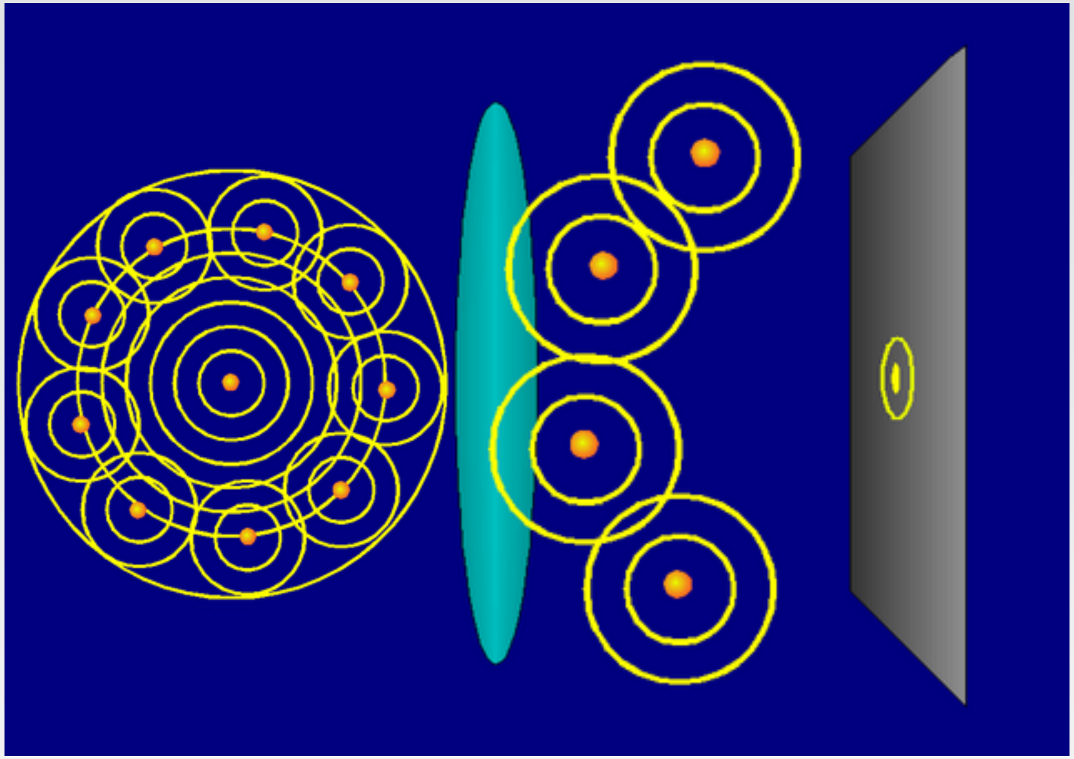


deconvolution

The goal of deconvolution is to find f given the kernel g and the output signal h

Source:

Deconvolution: Why we deconvolve?

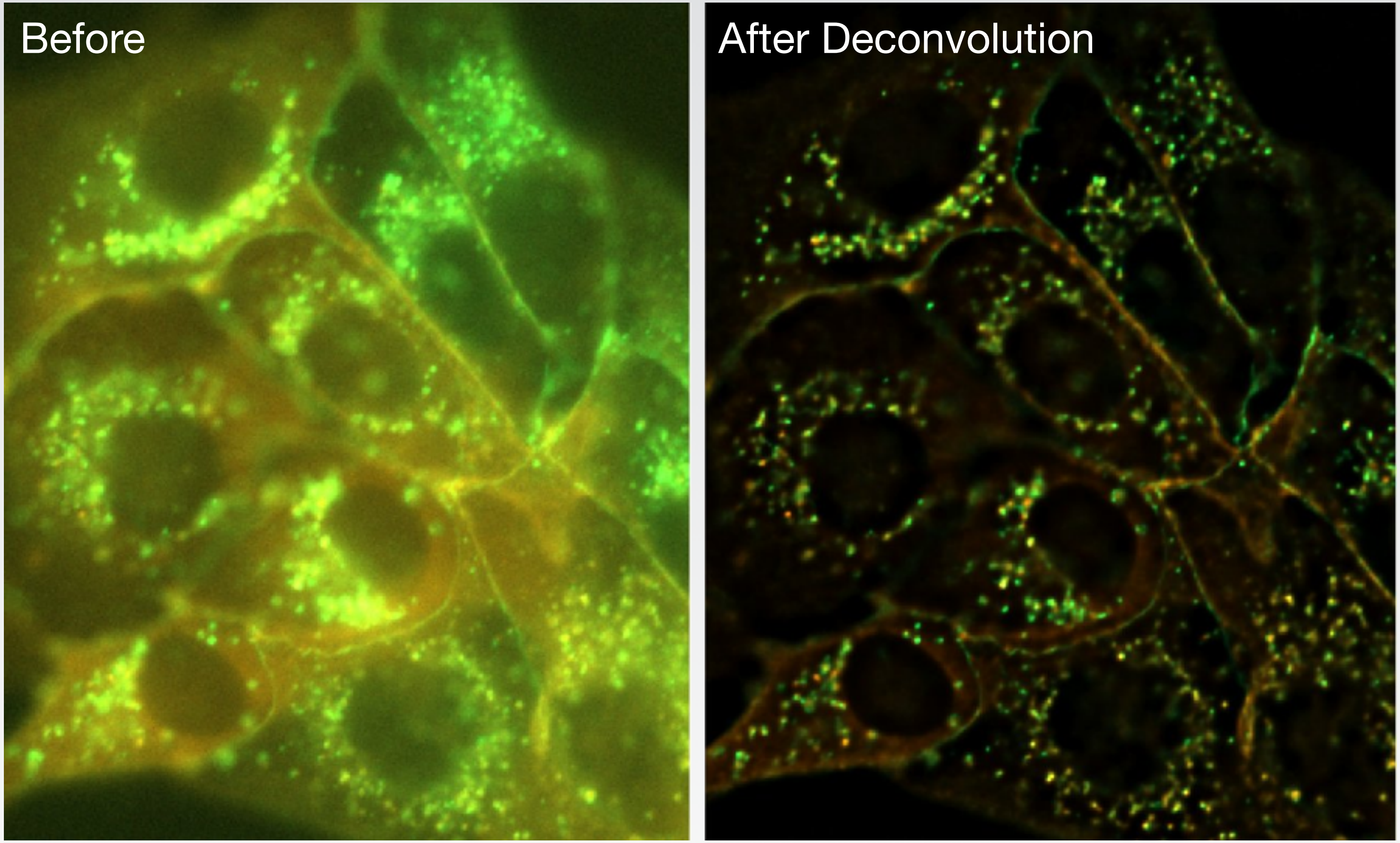


We have out-of-focus light contributing to our image plane, but we also have an artefact of the collection process influencing our final image.

Deconvolution seeks to remedy this issues.

Source: www.huygens.org

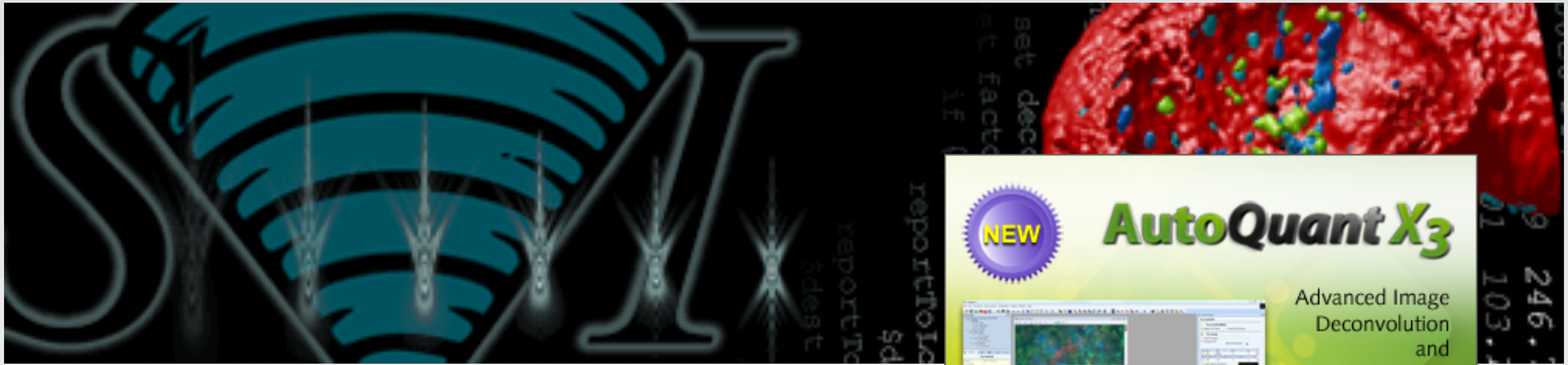
Huygens software



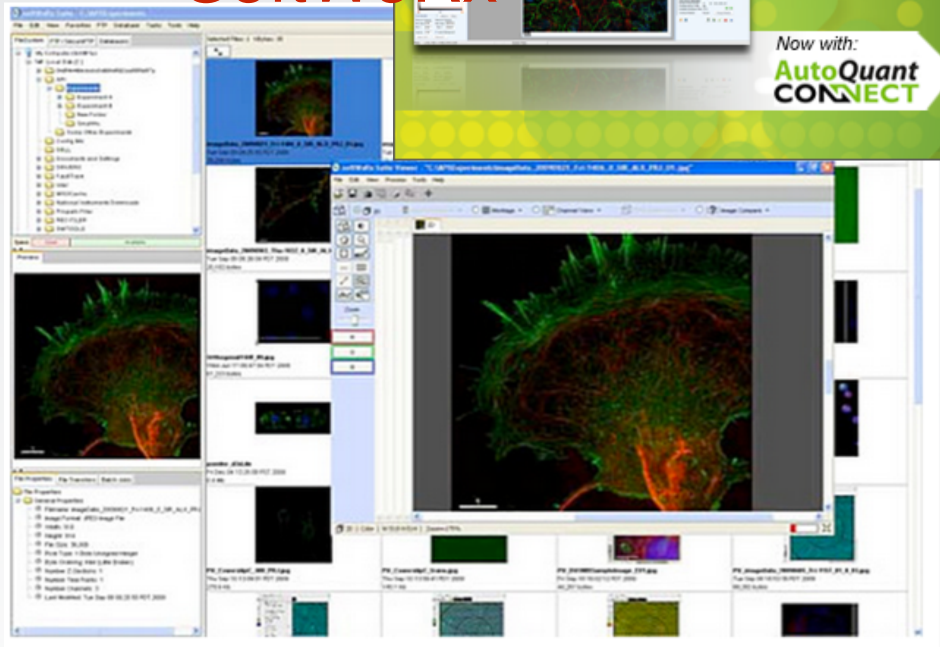
Deconvolution of cell-cell junctions of MDCK cells.
Nikon Ti widefield microscope (Objective 40x; 1.3 NA oil lens).

source: http://www.svi.nl/ImageGallery

Deconvolution software



SoftWoRx



Deconvolution

- Image restoration
- Volume Visualisation
- Some analysis

Expensive

WIMM CBRG has the 2nd fastest Huygens server in the world.

Source: http://www.svi.nl/HuygensSoftware, http://www.mediacy.com/index.aspx?page=AutoQuant, http://api.gehealthcare.com/api/softworx-suite.asp

Conclusions

- Image processing is fundamental to image analysis
- Filtering has many uses including noise removal
- Filtering can be linear or non-linear and play a role in many different types of processing.
- Using many filters in coordination allows texture analysis of objects and pixels.

Practical content and lectures

Doctoral Training Centre (MPLS) Resources /Modules/ 2016-17 / Michaelmas term / Week 9-10: Foundations of Image Analysis / day01_practicals

Summary of course: Course-schedule-overview.html

Practicals: day01_practicals/day01-prac-overview.html

Source: