DTU Compute

Image Analysis

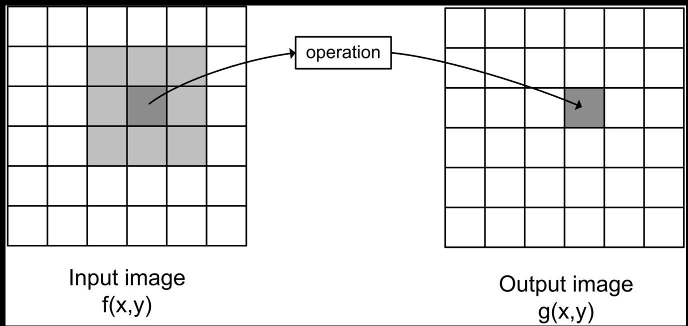
Tim B. Dyrby
Rasmus R. Paulsen
DTU Compute

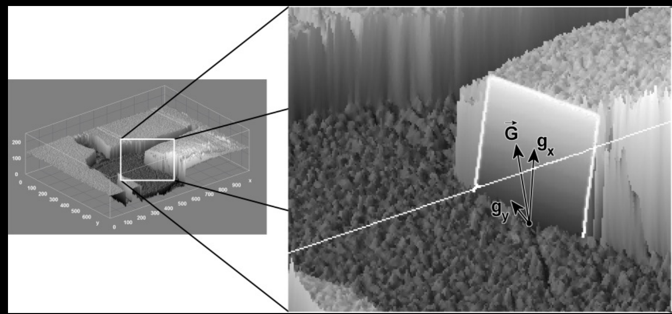
rapa@dtu.dk

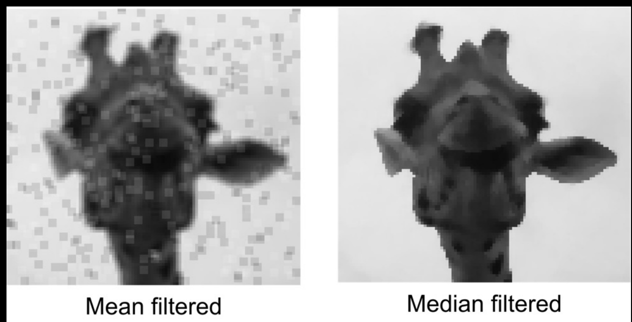
http://www.compute.dtu.dk/courses/02502

Plenty of slides adapted from Thomas Moeslunds lectures

Lecture 4
Neighbourhood Processing



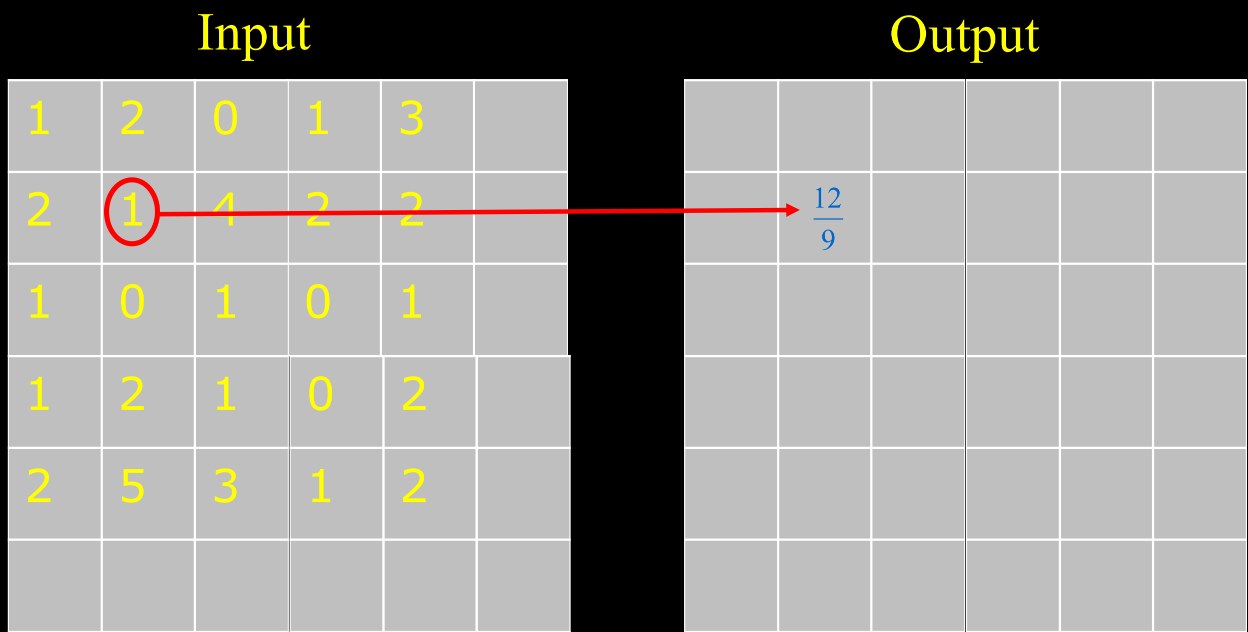




What can you do after today?

- Describe the difference between point processing and neighbourhood processing
- Compute a rank filtered image using the min, max, median, and difference rank filters
- Compute a mean filtered image
- Decide if median or average filtering should be used for noise removal
- Choose the appropriate image border handling based on a given input image
- Implement and apply template matching
- Compute the normalised cross correlation and explain why it should be used
- Apply given image filter kernels to images
- Use edge filters on images
- Describe finite difference approximation of image gradients including the magnitude and the direction
- Compute the magnitude of the gradient
- Compute edge detection in images

Point processing

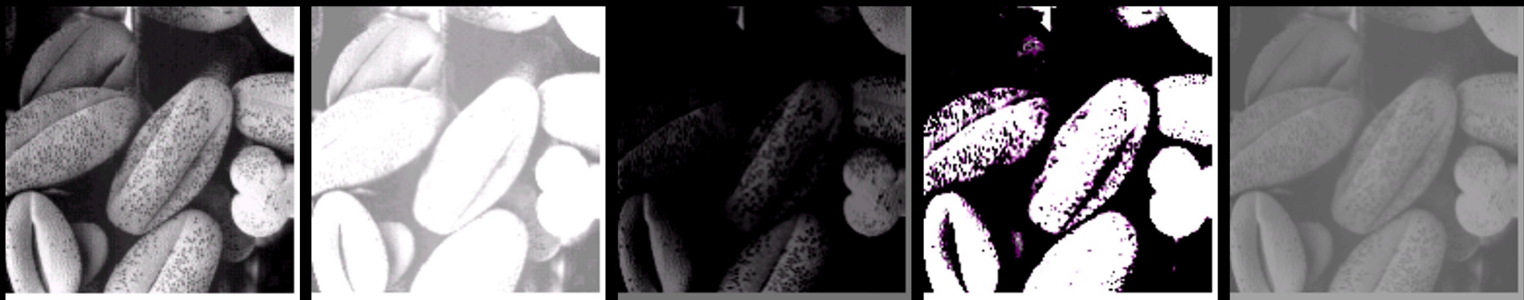


- The value of the output pixel is only dependent on the value of one input pixel
- A global operation – changes all pixels

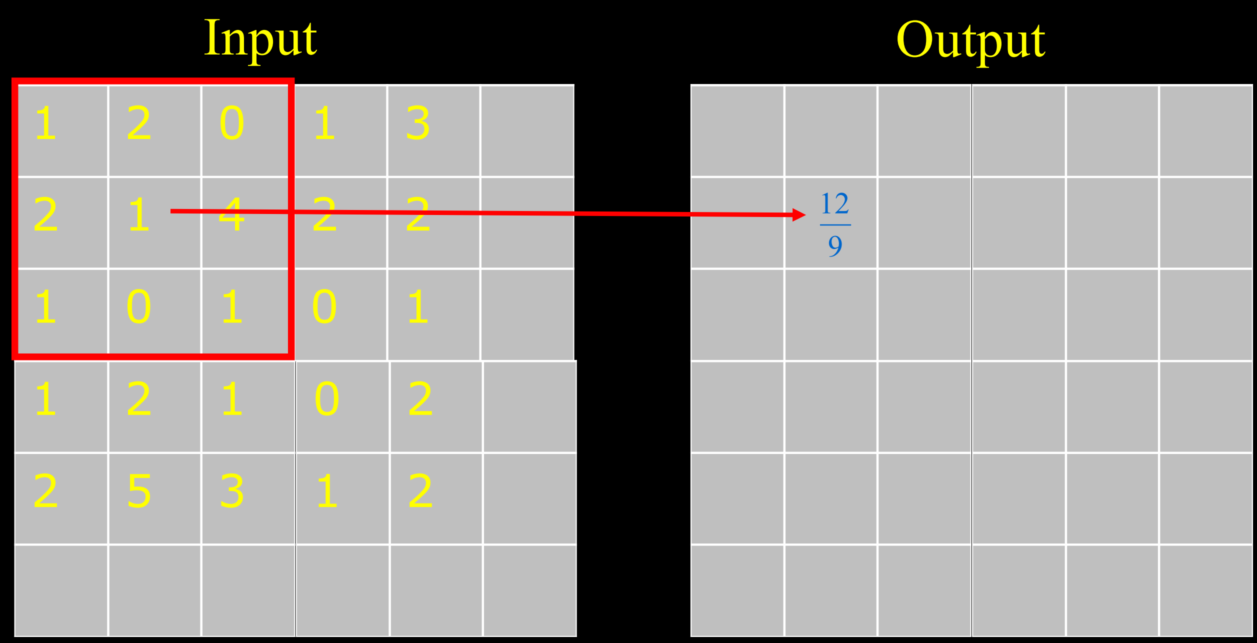
Point processing

- Grey level enhancement
  - Process one pixel at a time independent of all other pixels
  - For example, used to correct Brightness and Contrast

Correct | Too high brightness | Too low brightness | Too high contrast | Too low contrast



Neighbourhood processing



- Several pixels in the input has an effect on the output

Use of filtering

Noise removal

Enhance edges

Smoothing

- Image processing
- Typically done before actual image analysis

Salt and pepper noise

- Pixel values that are very different from their neighbours
- Very bright or very dark spots
- Scratches in X-rays

What is that?

Salt and pepper noise

- Fake example
  - Let us take a closer look at noise pixels

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

They are all 0 or 255

Should we just remove all the 0's and 255's from the image?

What is so special about noise?

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

172, 169, 171, 168, 0, 169, 172, 173, 168

- What is the value of the pixel compared to the neighbours?
- Average of the neighbours
  - 170
- Can we compare to the average?
  - Difficult – should we remove all values bigger than average+1 ?
- It is difficult to detect noise!

Noise – go away!

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

151

151

172, 169, 171, 168, 0, 169, 172, 173, 168

169, 168, 0, 170, 172,173, 170, 172, 170

- We cannot tell what pixels are noise
- One solution
  - Set all pixels to the average or mean of the neighbours (and the pixel itself)
- Oh no!
  - Problems!
  - The noise “pollutes” the good pixels

Quiz 1: What is the median value of [169, 168, 0, 170, 172, 173, 170, 172, 170]?

A) 170
B) 173
C) 169
D) 171
E) 172

The median value

- The values are sorted from low to high
- The middle number is picked
  - The median value

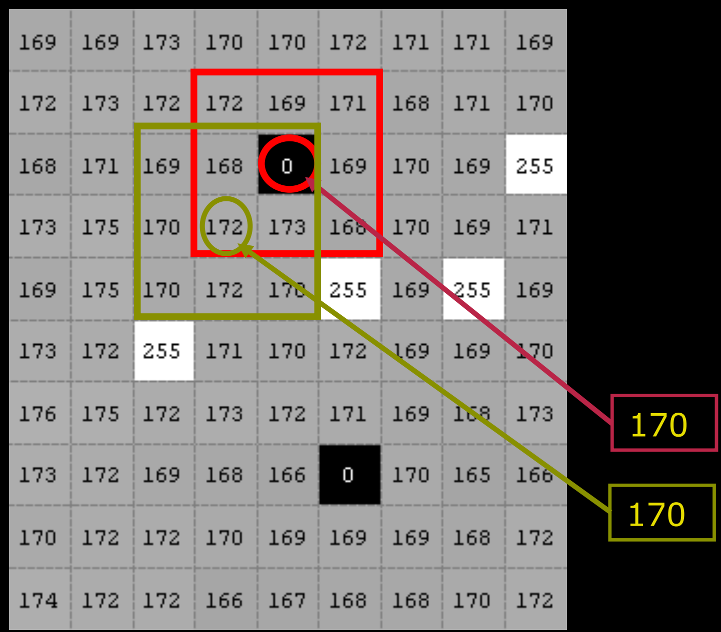
169, 168, 0, 170,172, 173, 170, 172, 170

0,168, 169, 170, 170, 170,172, 172,173

Median

Noise has no influence on the median!

Noise away – the median filter



172, 169, 171, 168, 0, 169, 172, 173, 168

169, 168, 0, 170, 172,173, 170, 172, 170

- All pixels are set to the median of its neighbourhood (including the pixel itself)
- Noise pixels do not pollute good pixels

Noise removal – average filter

Scanned X-ray with salt and pepper noise

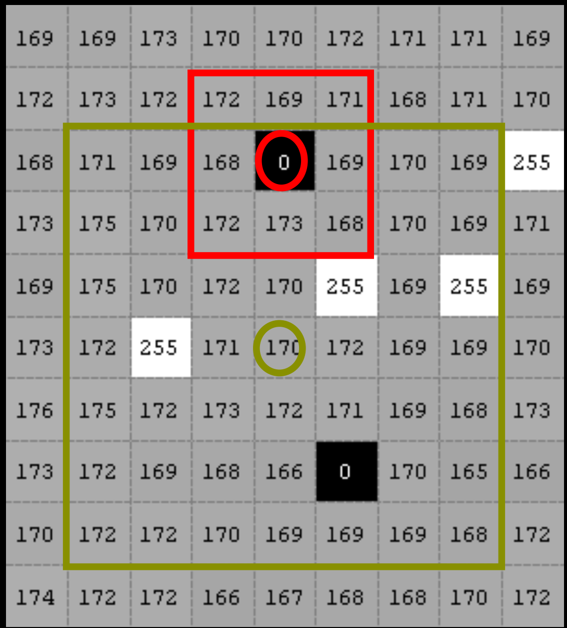
Average filter (3x3)

Noise removal – median filter

Scanned X-ray with salt and pepper noise

Median filter (3x3)

Image Filtering



- Creates a new *filtered* image
- Output pixel is computed based on a neighbourhood in the input image
- 3 x 3 neighbourhood
  - Filter size 3 x 3
  - Kernel size 3 x 3
  - Mask size 3 x 3
- Larger filters often used
  - Size
    - 7 x 7
  - Number of elements
    - 49

Quiz 2: Median filter on image

A) 25

B) 90

C) 198

D) 86

E) 103

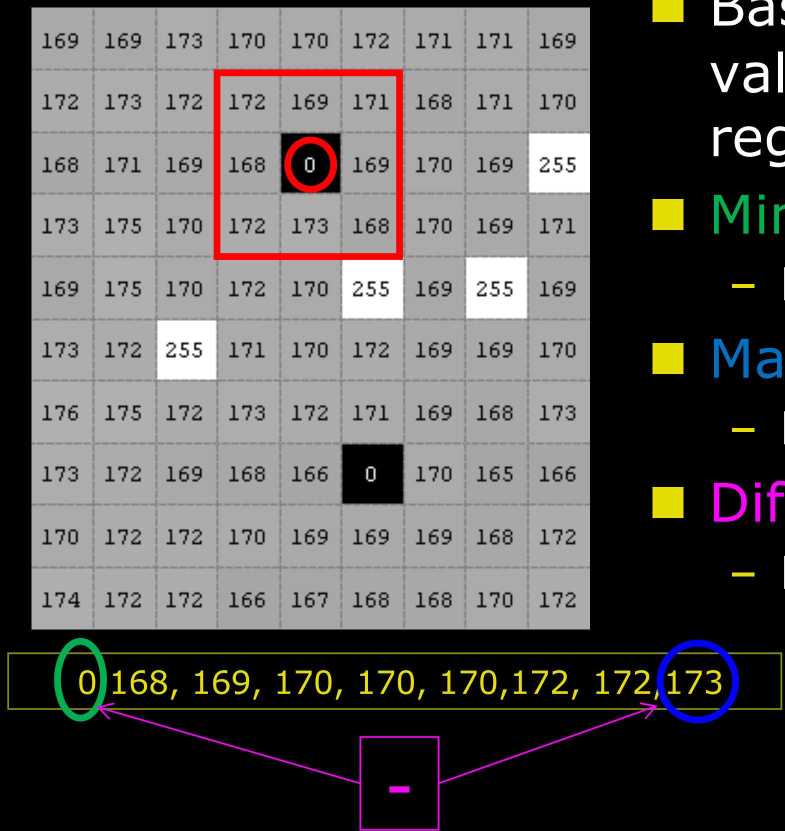
The image is filtered with a 3 x 3 median filter. What is the result in the pixel marked with a circle?

66	222	102	230	199	147	166	175
204	148	19	241	99	15	187	47
110	140	61	125	62	60	165	94
232	37	31	125	103	90	115	160
46	218	47	86	25	209	139	199
67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198

Answer:

4, 25, 34, 86, 90, 103, 125, 209, 230

Rank filters



- Based on sorting the pixel values in the neighbouring region as the median filter
- Minimum rank filter
  - Darker image. Noise problems.
- Maximum rank filter
  - Lighter image. Noise problems.
- Difference filter
  - Enhances changes (edges)

Quiz 3: Rank filters on image

The image is filtered with a 3 x 3 median filter (medI). The image (the original) is also filtered with a 5 x 5 minimum rank filter (minI). The final image is made by subtracting minI from medI. What is the result in the marked pixel?

67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198
222	102	230	199	147	166	175	124
148	19	241	99	15	187	47	111
140	61	125	62	60	165	94	114
37	31	125	103	90	115	160	78
218	47	86	25	209	139	199	130

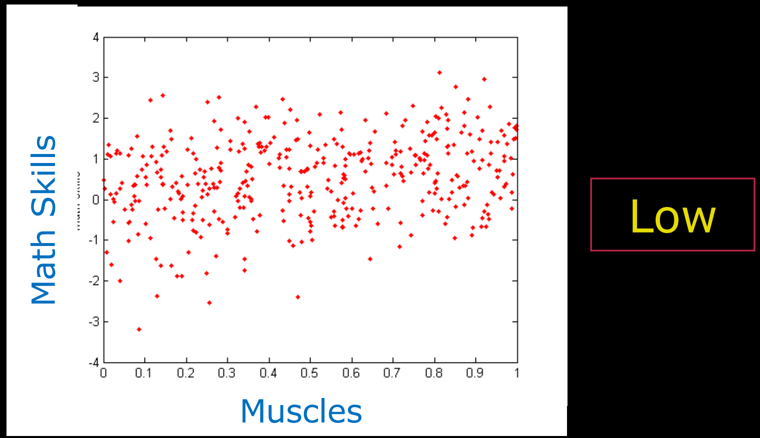
A) 3
B) 84
C) 112
D) 73
E) 202

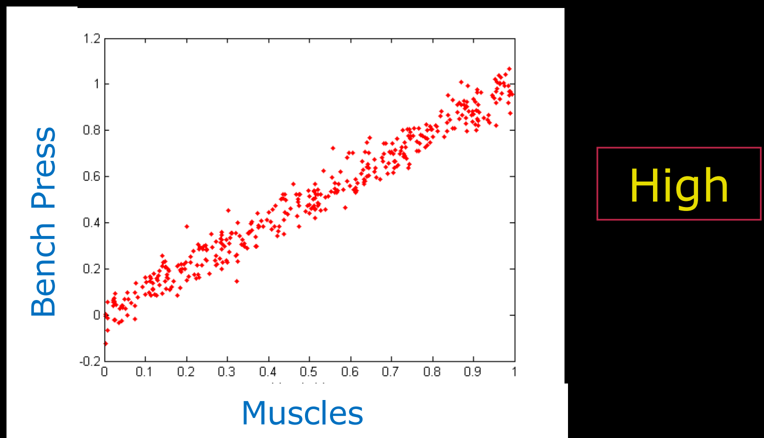
Answer:

medI: 15, 60, 62, 90, 99, 103, 115, 165, 187

minI: 15

Correlation

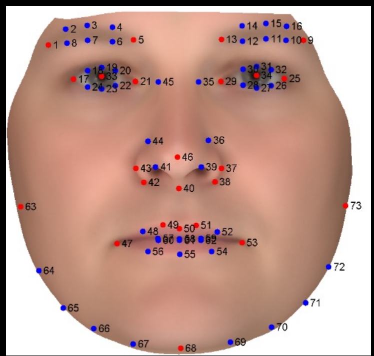




- What is it?
- Two measurements
  - Low correlation
  - High correlation
- High correlation means that there is a *relation* between the values
- They *look* the same
- Correlation is a *measure of similarity*

Why do we need similarity?

- Image analysis is also about recognition of patterns
- Often an example pattern is used
  - Often with some kind of meta data to apply to the targets
- We need something to tell us if there is a high match between our pattern and a part of the image

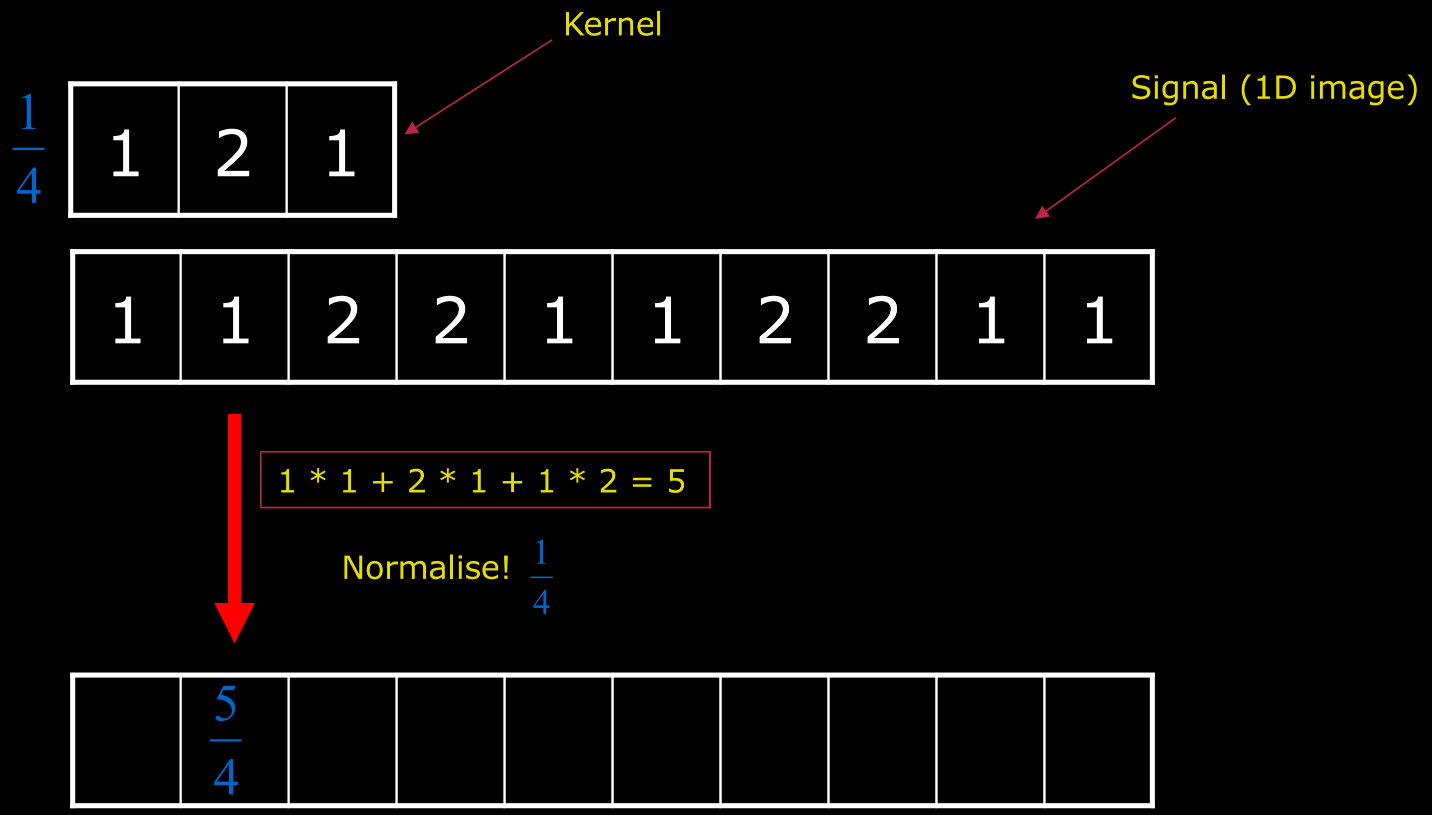


Find matches

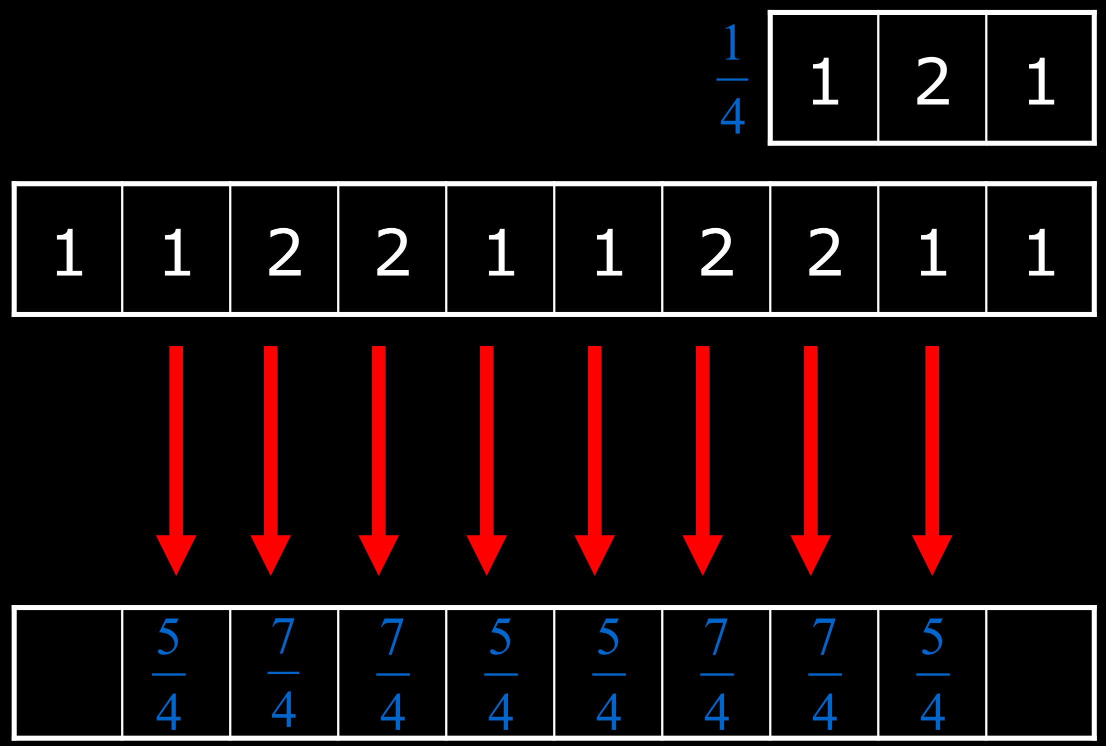
Example pattern

With meta information

Correlation (1D)



Correlation (1D)



Normalisation

- The sum of the kernel elements is used
- Keep the values in the same range as the input image

$\frac{1}{4}$

1	2	1

Sum is 4

1	1	2	2	1	1	2	2	1	1

1 * 1 + 2 * 1 + 1 * 2 = 5

Normalise! $\frac{1}{4}$

	$\frac{5}{4}$								

Normalisation

- Normalisation factor
  - Sum of kernel coefficients

h(x)

1	2	1

$$\sum_x h(x) = 1 + 2 + 1$$

Correlation on images

- The filter is now 2D

Kernel coefficients

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Kernel

Input

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Output

	$\frac{12}{9}$				

Correlation on images

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

Input

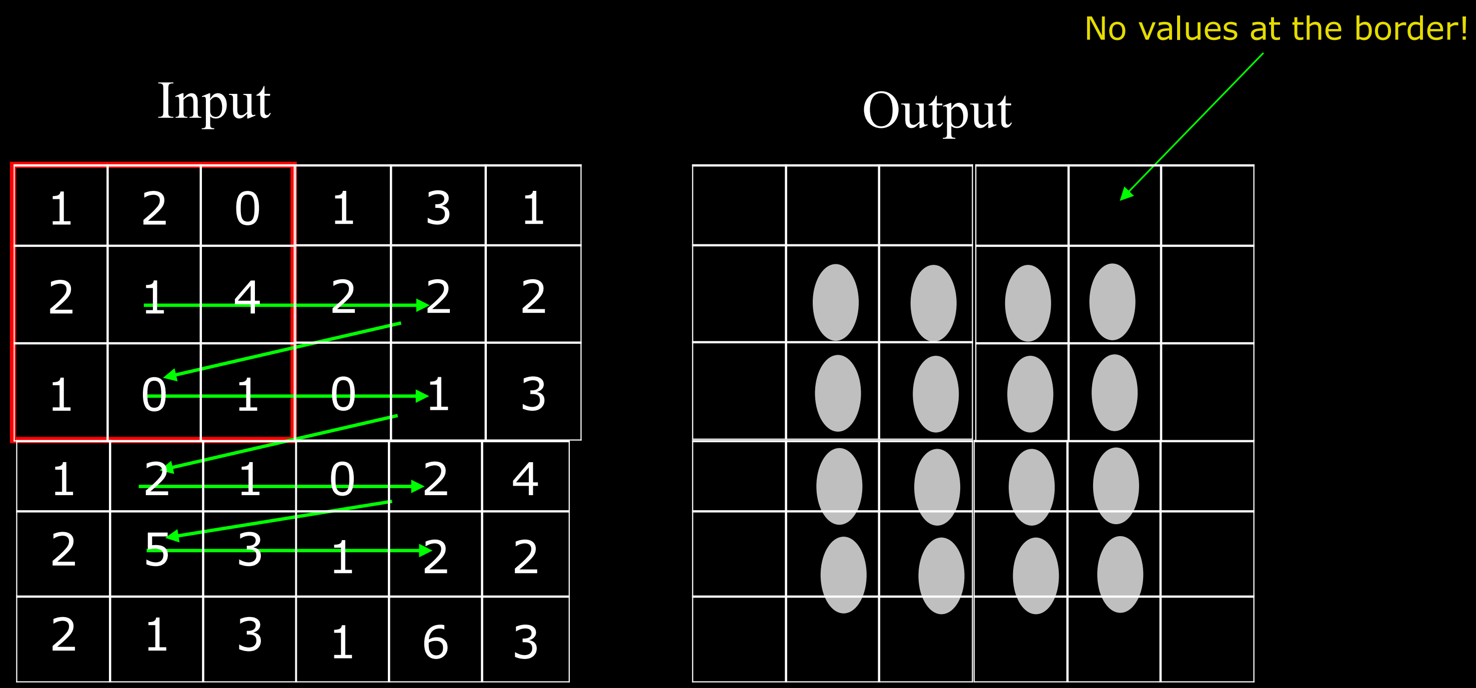
1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Output

	$\frac{12}{9}$	$\frac{11}{9}$			

Correlation on images

The mask is moved row by row



Quiz 4: Correlation on image – no normalisation

A) 4
B) 7
C) 16
D) 23
E) 25

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Quiz 5: Correlation on image 2 – no normalisation

A) 0,10
B) 3, 3
C) 6, 2
D) 10, 15
E) 11, 32

-1	-2	-1
0	0	0
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Mathematics of 2D Correlation

$$g(x,y) = f(x,y) \circ h(x,y)$$

Correlation operator

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

f

1	2	1
1	3	1
1	2	1

h

Mathematics of 2D Correlation

$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

Example g(2,1)

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

f

f(2,1)

1	2	1
1	3	1
1	2	1

h

h(0,0)

i = -1, j = -1

i = 1, j = 0

Mathematics of 2D Correlation

$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

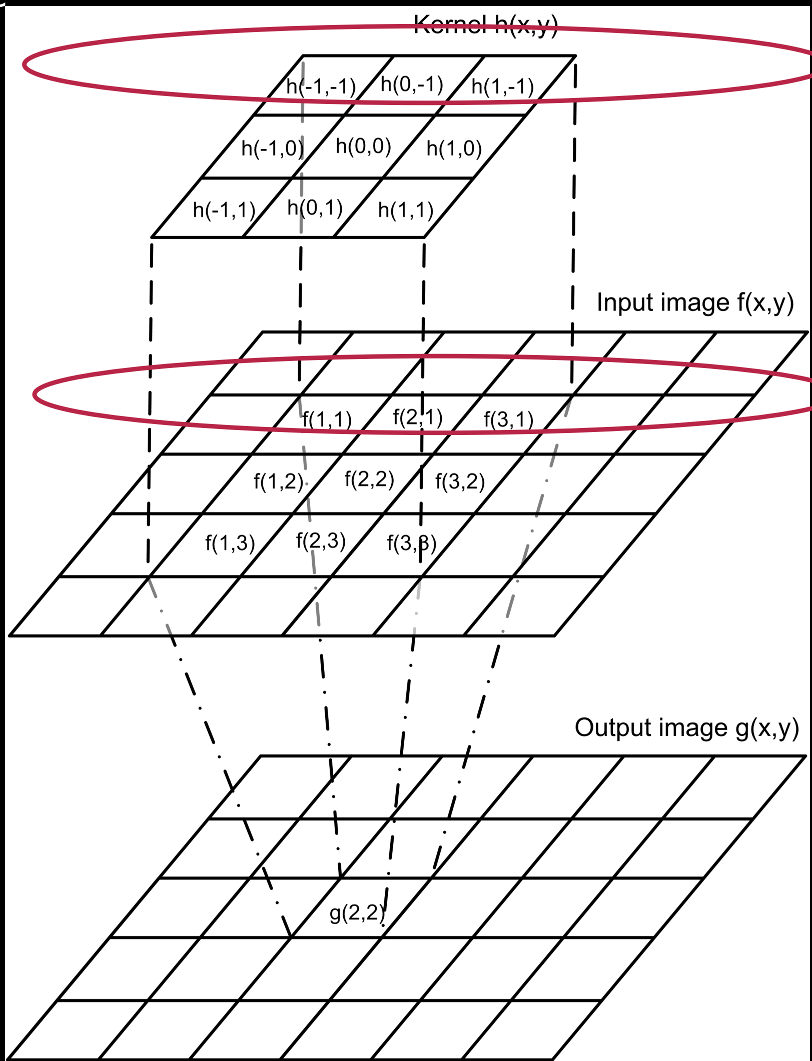
$$g(x,y) = 1 \cdot 2 + 2 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 + 3 \cdot 4 + 1 \cdot 2 + 1 \cdot 0 + 2 \cdot 1 + 1 \cdot 0$$

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

f

1	2	1
1	3	1
1	2	1

h



$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

$$g(2,2) =$$
$$h(-1,-1) \cdot f(1,1) + h(0,-1) \cdot f(2,1) + h(1,-1) \cdot f(3,1) +$$
$$h(-1,0) \cdot f(1,2) + h(0,0) \cdot f(2,2) + h(1,0) \cdot f(3,2) +$$
$$h(-1,1) \cdot f(1,3) + h(0,1) \cdot f(2,3) + h(1,1) \cdot f(3,3)$$

Correlation is a sliding dot-product between two vectors

$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

- In correlation we multiply *h* and *f*
- Kernel *h* and 3x3 patch in figure *f* are two high dimensional vectors:

  $f(2,1) = [2, 0, 1, 1, 4, 2, 0, 1, 0]$

  $h = [1, 2, 1, 1, 3, 1, 1, 2, 1]$

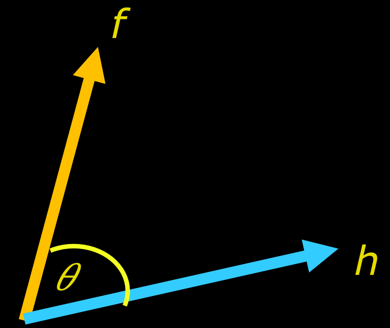
- Multiplication is a dot product
  - $h \cdot f = \|h\| \, \|f\| \cos\theta$

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

f

1	2	1
1	3	1
1	2	1

h



2D Kernel Normalisation

Normalisation factor:

$$\sum_{x}\sum_{y} h(x,y)$$

$$1+2+1+1+3+1+1+2+1=13$$

1	2	1
1	3	1
1	2	1

h

Quiz 6: Template match on image

A) 50122
B) 123001
C) 11233
D) 2550
E) 90454

A template match is done on the image to the left with the template seen to the right. To find the best match the correlation is computed. What is the correlation in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

66	232	37
204	46	35
110	67	222

Smoothing filters

- Also know as
  - Smoothing kernel, Mean filter, Low pass filter, blurring
- The simplest filter:
  - *Spatial low pass filter*
  - Removes high frequencies

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

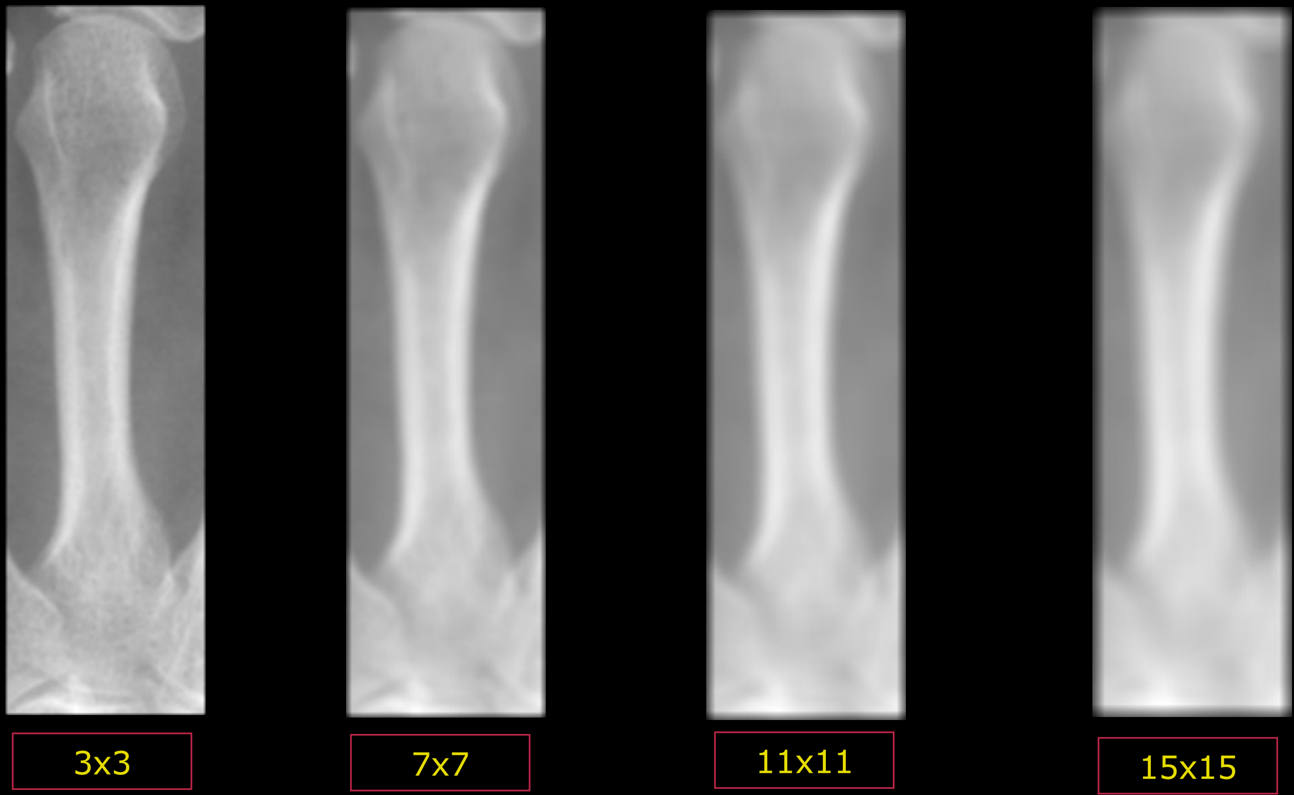
- Another mask:
  - Gaussian filter

$$\frac{1}{16}$$

1	2	1
2	4	2
1	2	1

Why Gaussian?

Use of smoothing



Use of smoothing

3x3

15x15

- Large kernels smooth more
- Removes high frequency information
- Good at enhancing *big structures*

Quiz 7: Mean filter on image

A) 166
B) 113
C) 12
D) 51
E) 245

A 3x3 mean filter is applied to the image. The result in the marked pixel is 86. What is the value of the pixel, where the value is missing?

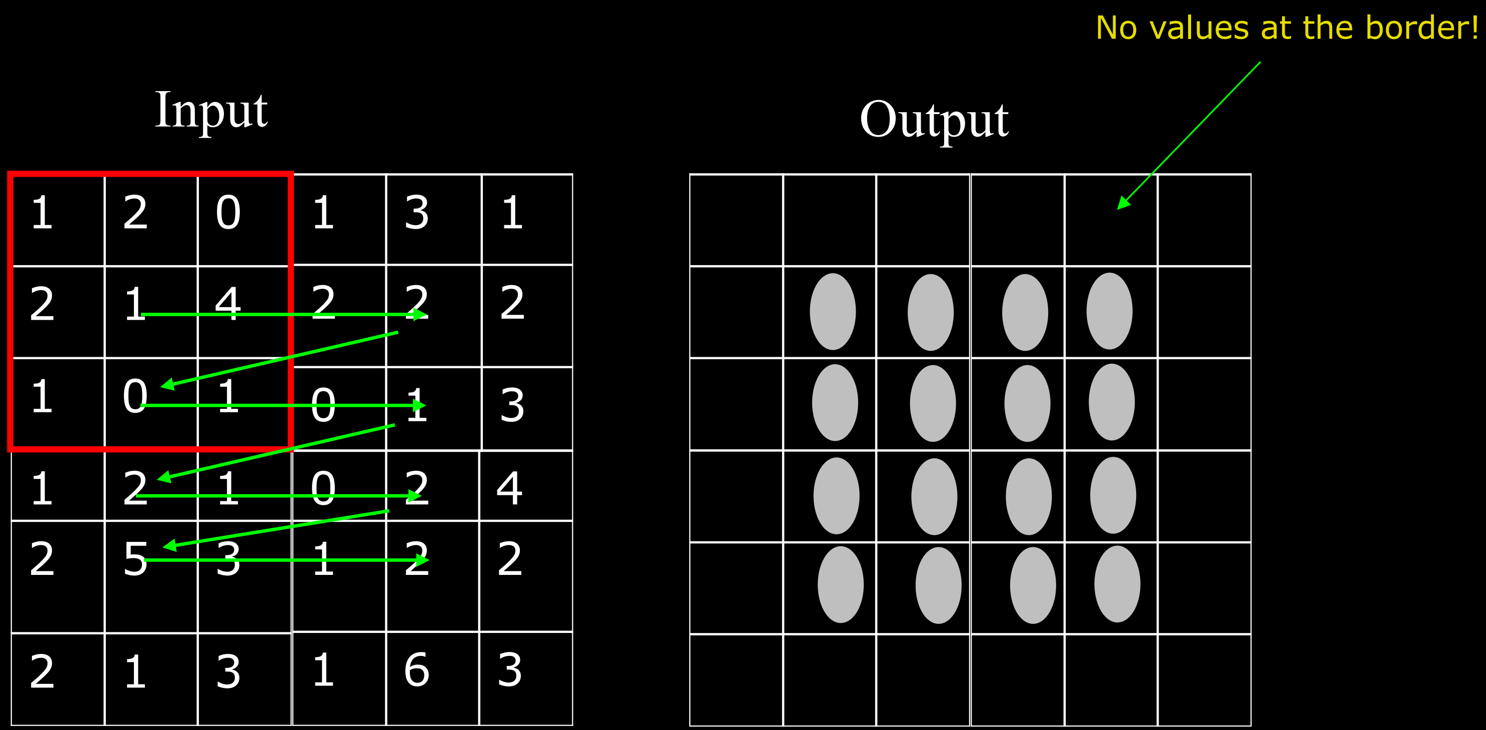
227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86		211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

Answer:

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

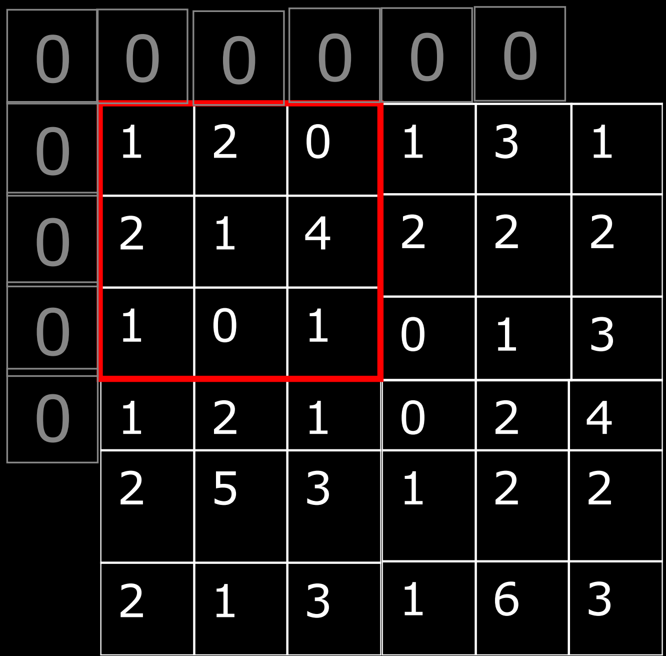
(608+x*1)/9=86 -> x=166

Border handling

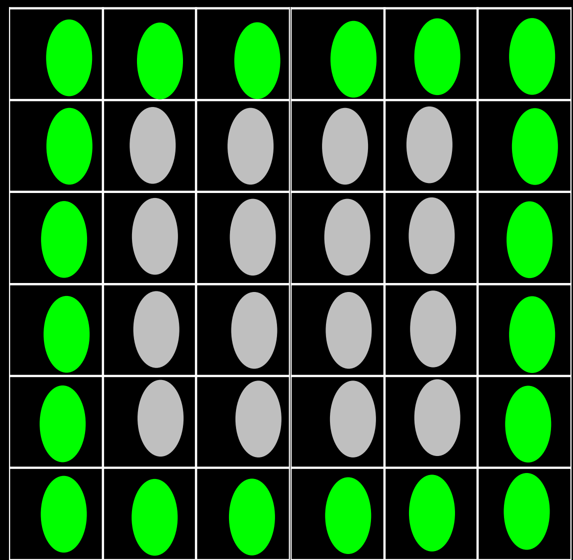


Border handling – extend the input

Input



- Zero padding – what happens?
- Zero is black – creates dark border around the image



Quiz 8: Correlation on image with zero padding

A) 5, 8
B) 12, 16
C) 13, 3
D) 15, 21
E) 17, 12

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Quiz 8: Correlation on image with zero padding

A) 5, 8
B) 12, 16
C) 13, 3
D) 15, 21
E) 17, 12

Answers:

1	2	1
1	3	1
1	2	1

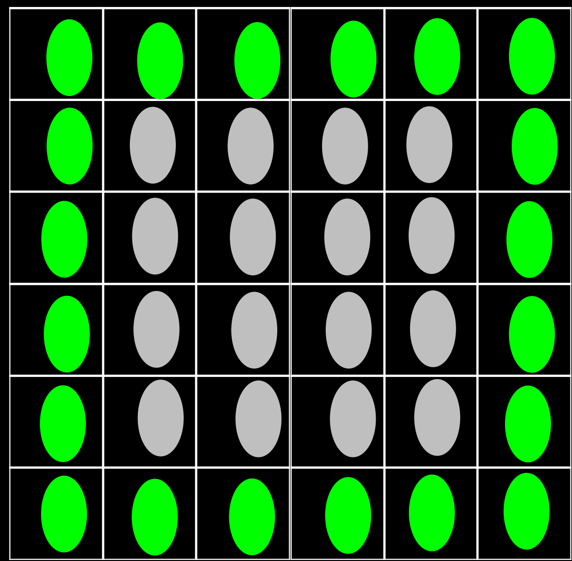
1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Border handling – extend the input

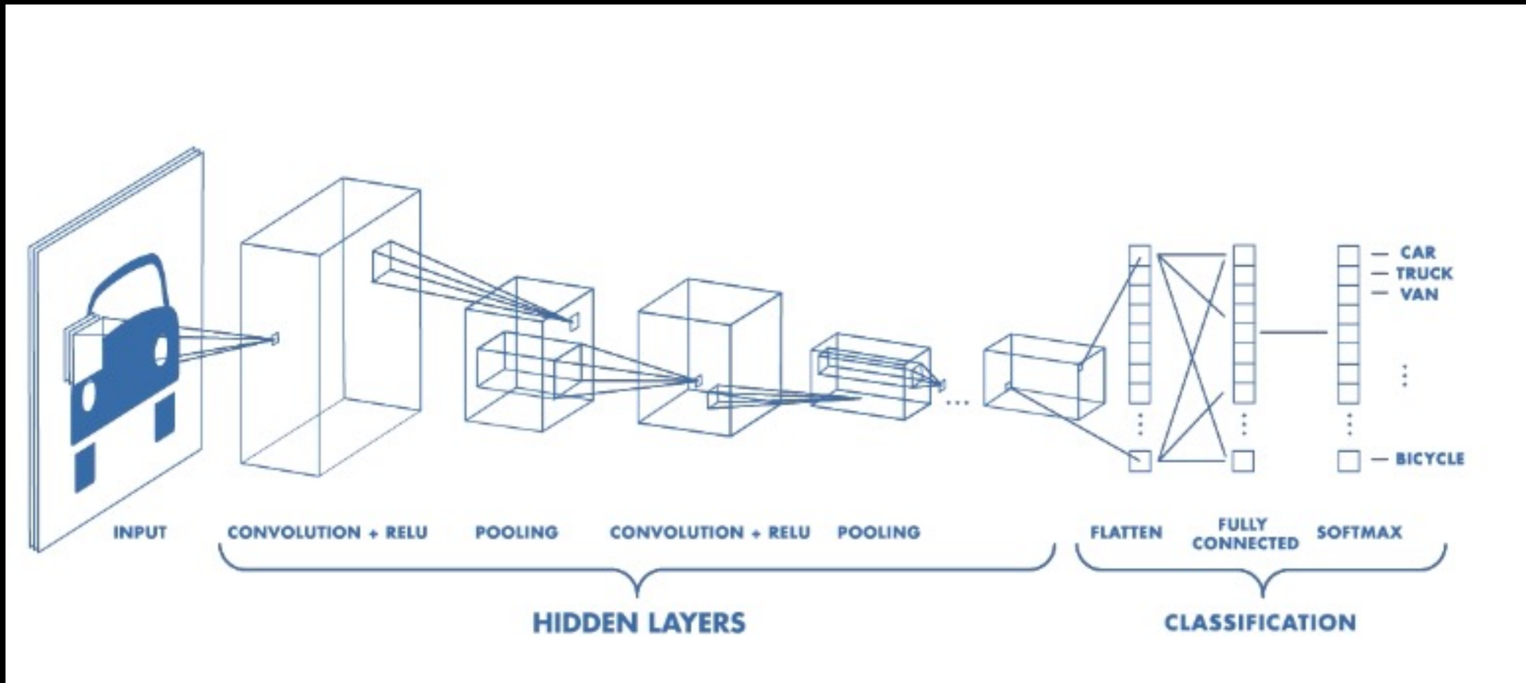
- Reflection
- Normally better than zero padding

Input

1	1	2	0	1	3	
1	1	2	0	1	3	1
2	2	1	4	2	2	2
1	1	0	1	0	1	3
1	1	2	1	0	2	4
	2	5	3	1	2	2
	2	1	3	1	6	3



What is the connection to deep learning?

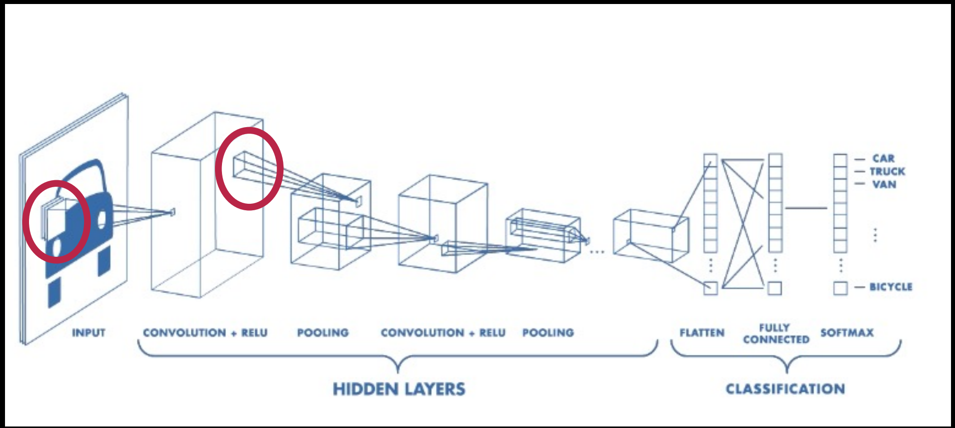


https://se.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

Banks of filters

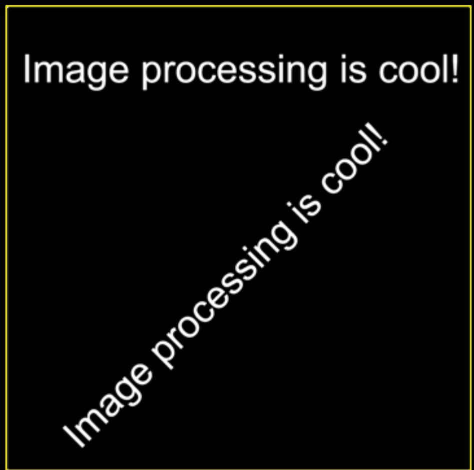
- The part of the network that touches the image consists of a bank of filters
  - Organised in a multi-level hierarchy

- The weights of the filters are adapted to the problem



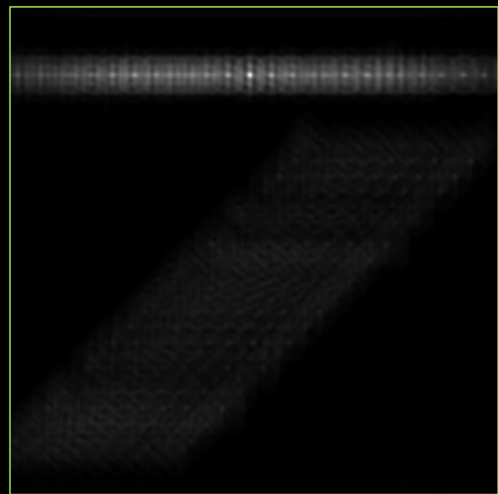
Template Matching

- Template
  - *Skabelon* på dansk
- Locates objects in images



Input

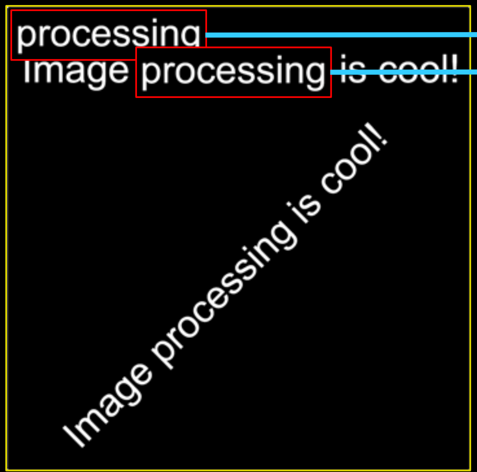
Template



Output: Correlation image

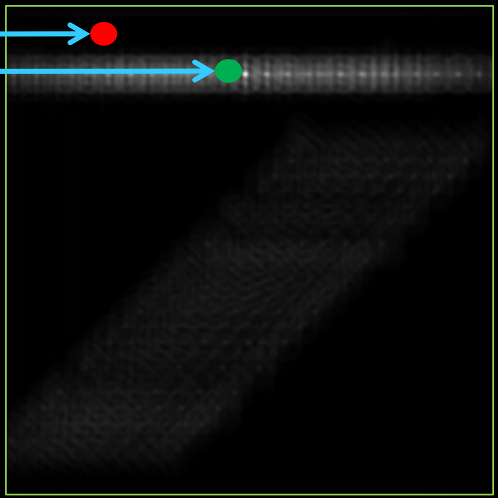
Template Matching

- The correlation between the template and the input image is computed for each pixel



Input

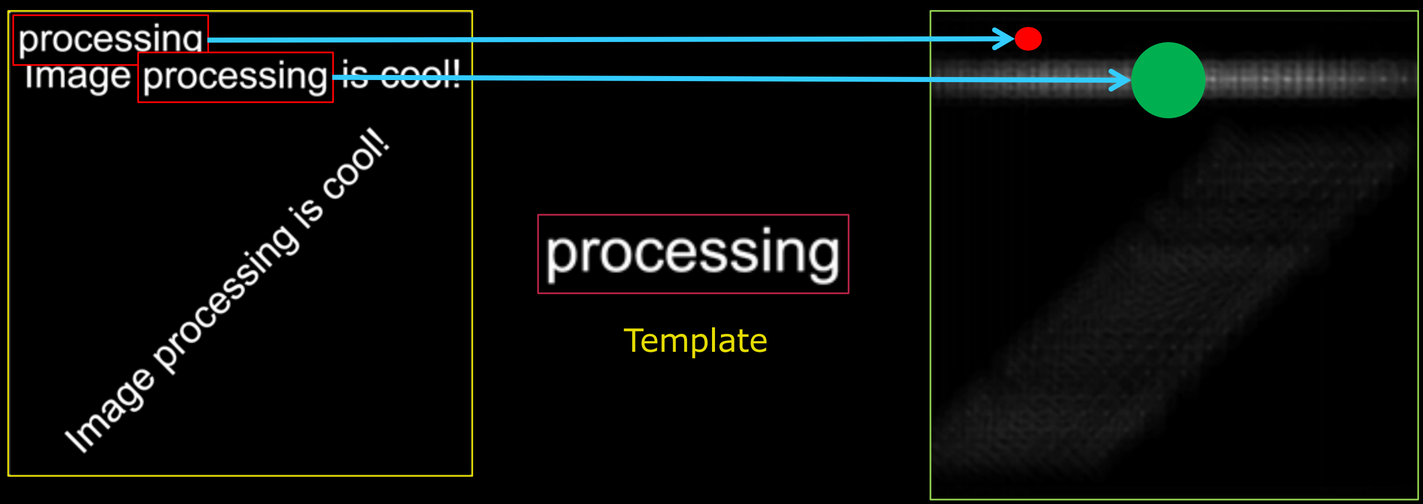
Template



Output: Correlation image

Template Matching

- The pixel with the highest value is found in the output image
  - Here is the highest correlation



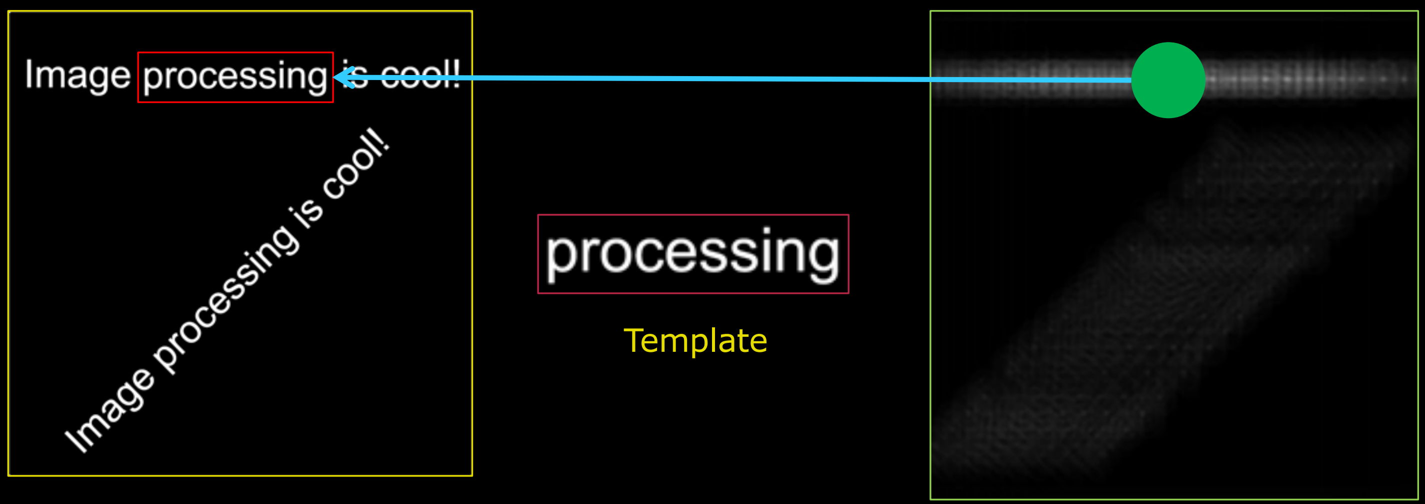
Input

Template

Output: Correlation image

Template Matching

- This corresponds to the found pattern in the input image



Input

Template

Output: Correlation image

Problematic Correlation

- Correlation matching has problem with light areas – why?

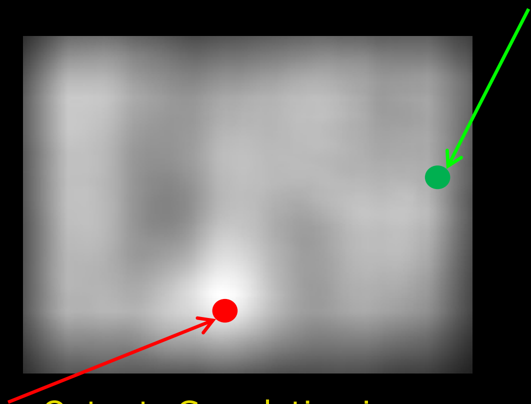
$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

Very High!

Input (f)

Template (h)

Real max



Fake max

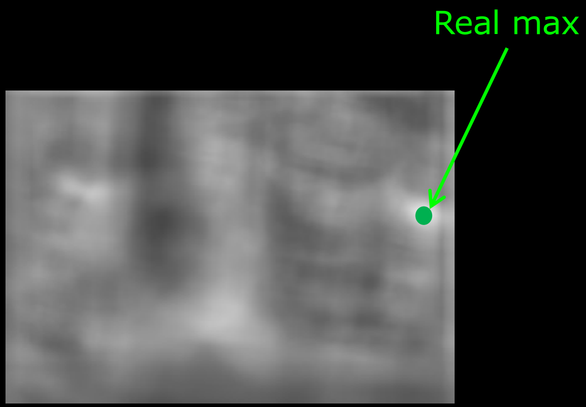
Output: Correlation image

Normalised Cross Correlation

$$\mathrm{NCC}(x,y) = \frac{\text{Correlation}}{\text{Length of image patch} \cdot \text{Length of template}}$$

Input (f)

Template (h)



Output: Correlation image

Length of template

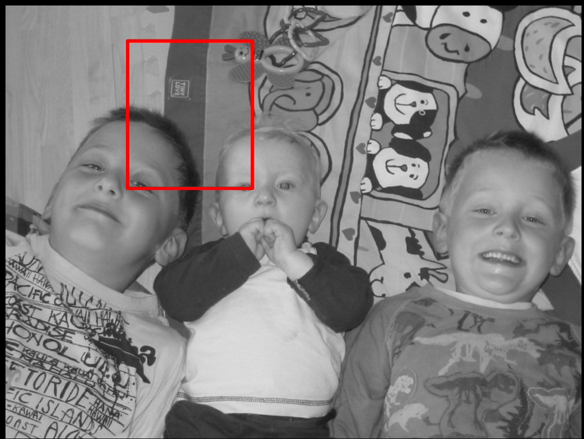
- Vector length
  - Put all pixel values into a vector
  - Compute the length of this vector
- Describes the intensity of the template
  - Bright template has a large length
  - Dark template has a small length

$$\text{Length of template} = \sqrt{\sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot h(i,j)}$$

Template (h)

Length of image patch

- Vector length based on pixel values in image patch
- Describes the intensity of the image patch



Input (f) with patch

Template (h)

Normalised Cross Correlation

- The length of the image patch and the length of template normalises the correlation
- Since correlation is a dot product:
- Normalising a dot product results in the angle between vector f and h

$$\text{NCC}(x, y) = \frac{\text{Correlation}}{\text{Length of image patch} \cdot \text{Length of template}}$$

- Normalised dot product

$$\text{NCC}(x,y) = \cos\theta$$

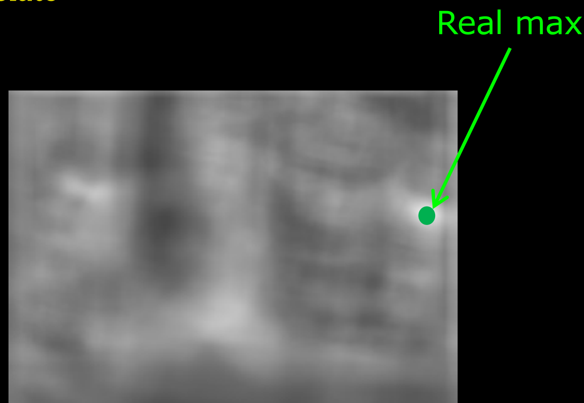
Normalised Cross Correlation

- NCC will be between
  - 0 : No similarity between template and image patch
  - 1 : Template and image patch are identical

$$NCC(x,y) = \frac{\text{Correlation}}{\text{Length of image patch} \cdot \text{Length of template}}$$

Input (f)

Template (h)



Output: Correlation image

Quiz 9: Normalised cross correlation on image

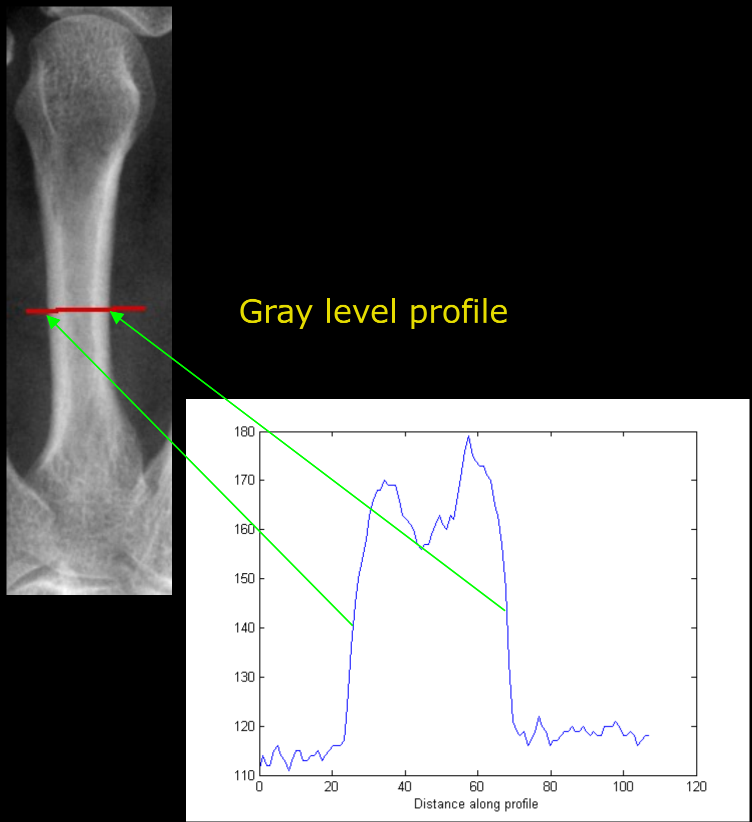
A template match using normalised cross correlation is performed. What is the resulting value in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

66	232	37
204	46	35
110	67	222

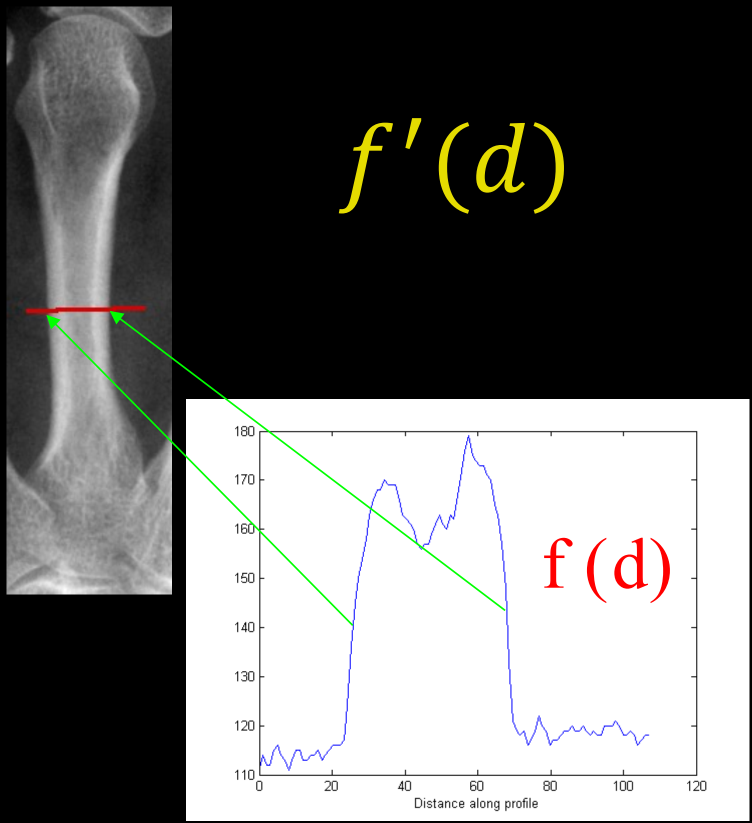
A) 0.10
B) 0.33
C) 0.83
D) 0.62
E) 0.98

Edges



- An edge is where there is a high change in gray level values
- Objects are often separated from the background by edges

Edges



$$f'(d)$$

- The profile as a function f(d)
- What value is high when there is an edge?
  - The slope of f
  - The slope of the tangent at d

Finite Difference

- Definition of slope

$$f'(d) = \lim_{h \to 0} \frac{f(d+h) - f(d)}{h}$$

- Approximation

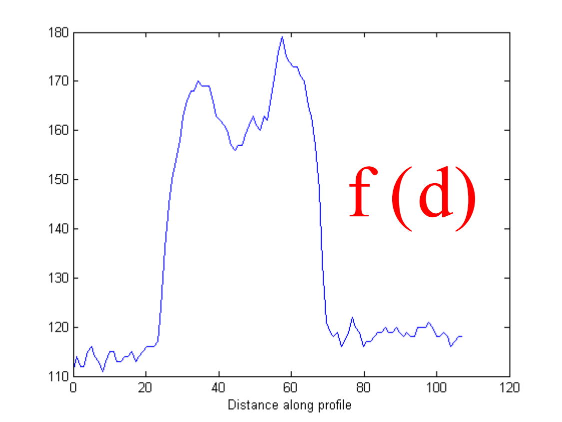
$$f'(d) \approx \frac{f(d+h) - f(d)}{h}$$

- Simpler approximation

$$\mathrm{h} = 1$$

$$f'(d) \approx f(d+1) - f(d)$$

Edges



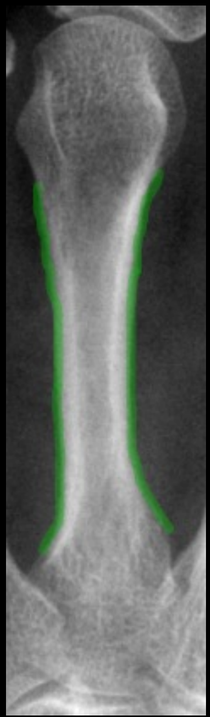
- Discrete approximation of f'(d)
- Central differences is used: d=+/-1
- Can be implemented as a filter

$$f'(d) \approx f(d \pm 1) - f(d)$$

-1	0	1

f(d-2)	f(d-1)	f(d)	f(d+1)	f(d+2)	f(d+3)

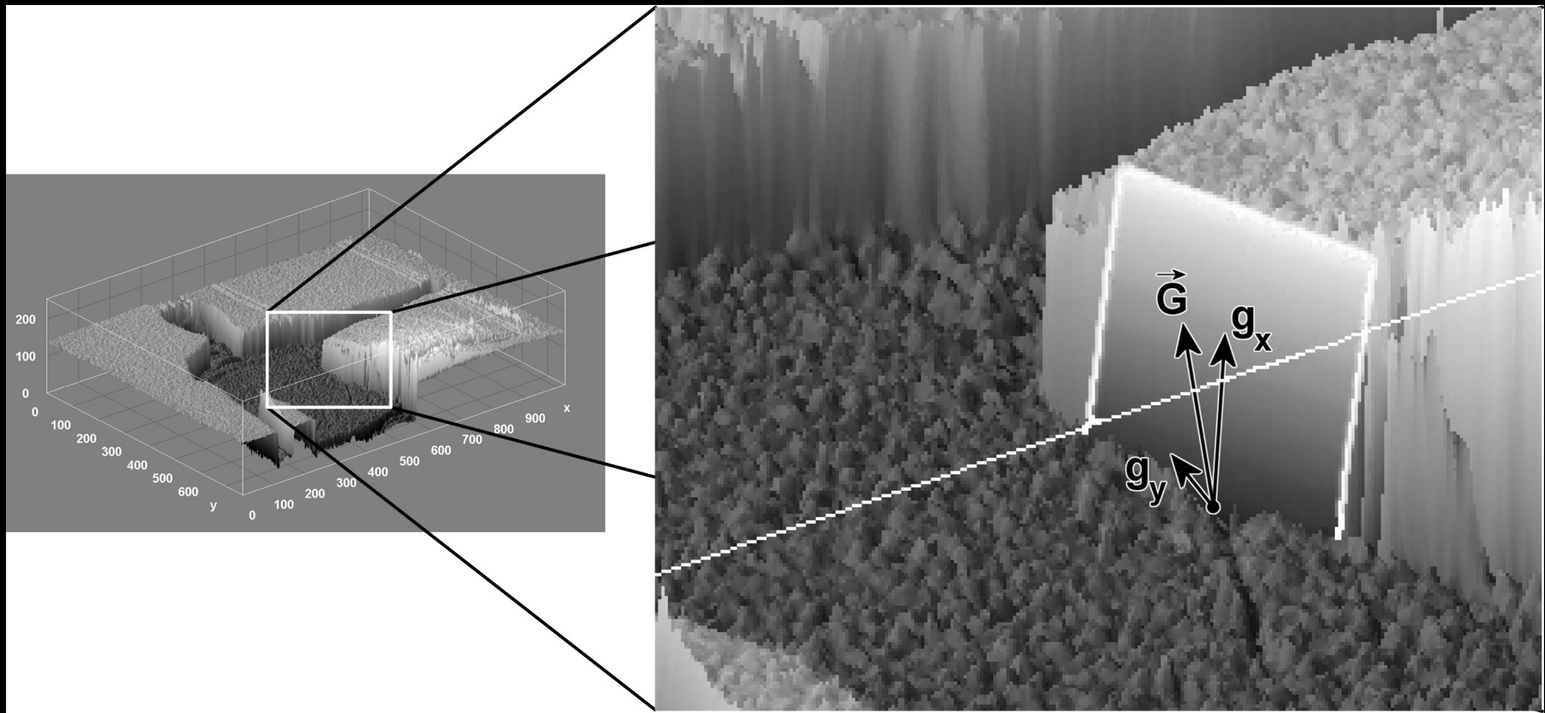
Edges in 2D



- Changes in gray level values
  - Image gradient
  - Gradient is the 2D derivative of a 2D function f(x,y)
  - Equal to the *slope* of the image
  - A steep slope is equal to an edge

$$\nabla f(x,y) = \vec{G}(g_x, g_y)$$

2D Gradient



$$\text{magnitude} = \sqrt{g_x^2 + g_y^2}$$

Edge filter kernel

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

Vertical Prewitt filter

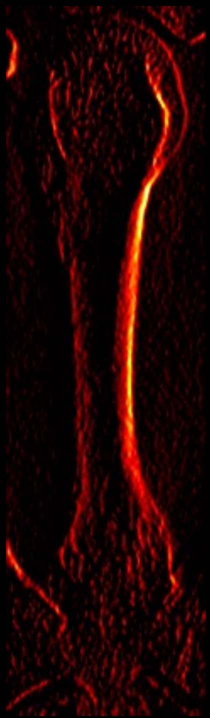
(gradient: g(x))

- The Prewitt filter is a typical edge filter
- Output image has high values where there are edges

Prewitt filter (vertical)

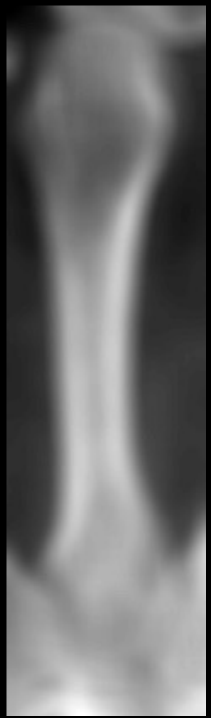
Original

Prewitt

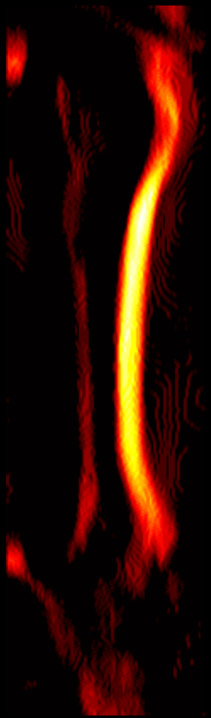


Prewitt

Hot colormap



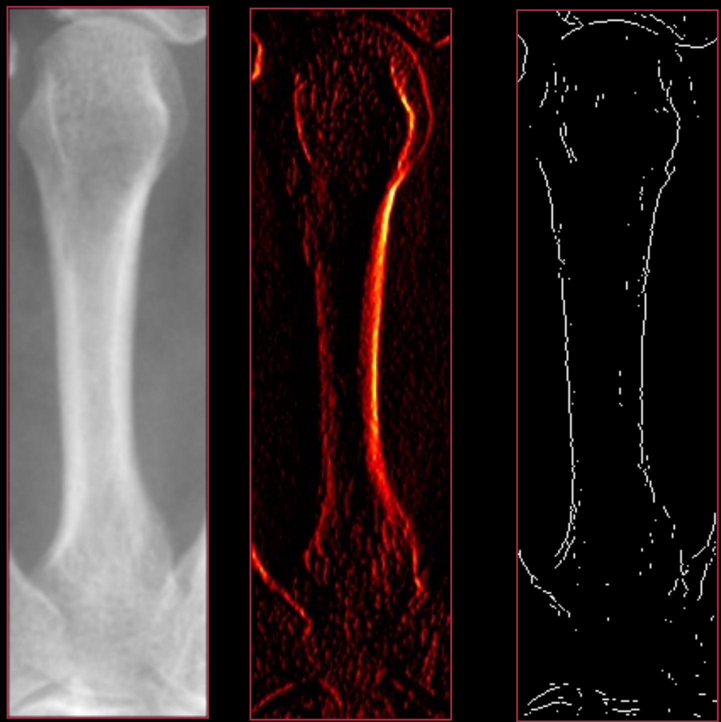
Smooth 15x15



Smooth 15x15

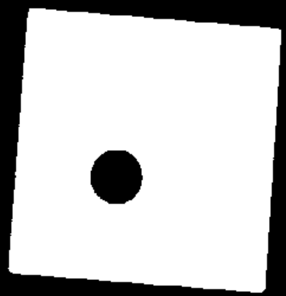
Prewitt

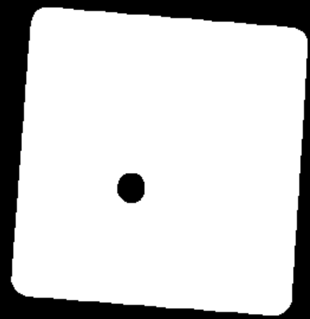
Edge detection



- Edge filter
  - Prewitt for example
- Find magnitude of 2D gradient
- Thresholding
  - Separate edges from non-edges
- Output is binary image
  - Edges are white

Lecture 4b – Morphology





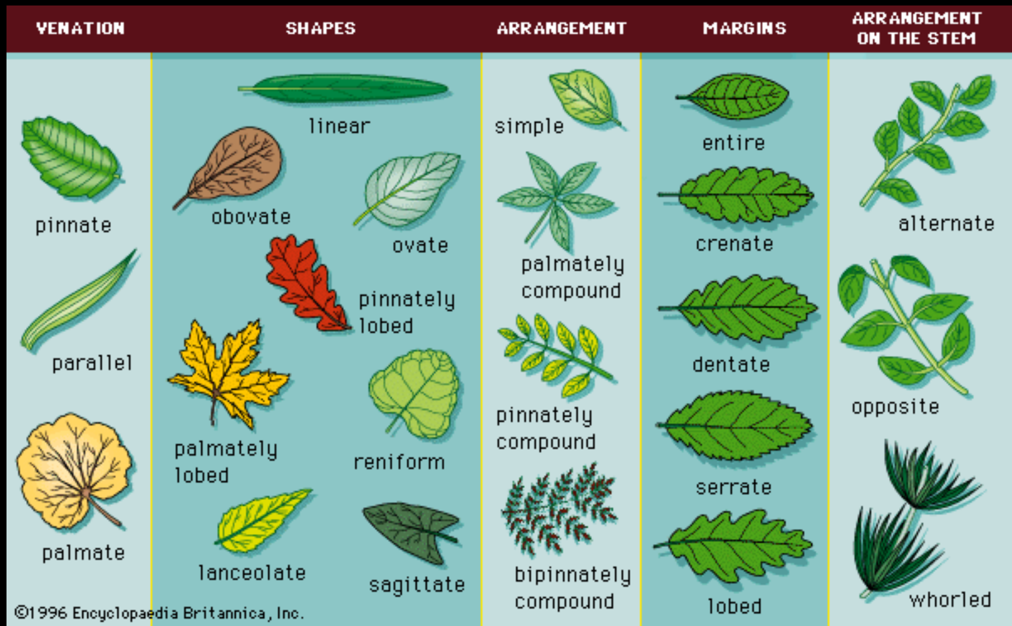
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0

What can you *also* do after today?

- Describe the similarity between filtering and morphology
- Describe a structuring element
- Compute the dilation of a binary image
- Compute the erosion of a binary image
- Compute the opening of a binary image
- Compute the closing of a binary image
- Apply compound morphological operations to binary images
- Describe typical examples where morphology is suitable
- Remove unwanted elements from binary images using morphology
- Choose appropriate structuring elements and morphological operations based on image content

Morphology

- The science of *form, shape* and *structure*
- In biology: The form and structure of animals and plants



Common leaf morphologies

Mathematical morphology

Theorem 4.10
$$\begin{cases} \psi_m = \tilde{\varphi}\,\tilde{\gamma} = \tilde{\gamma}\,\tilde{\varphi}\,\tilde{\gamma} = \psi\tilde{\gamma} & , \\ \psi_M = \tilde{\gamma}\tilde{\varphi} = \tilde{\varphi}\,\tilde{\gamma}\,\tilde{\varphi} = \psi\tilde{\varphi} & , \\ \psi = \tilde{\gamma}\,\psi = \tilde{\varphi}\,\psi, & \\ \tilde{\gamma} \leq \psi_m \leq \psi \leq \psi_M \leq \tilde{\varphi} & . \end{cases}$$

The same theorem may be restated in another way. If $\mathcal{J}\mathrm{d}(\mathcal{B}) \neq \varnothing$ then let B_i be a family of elements of $\mathcal{B}$. We have $\vee\, B_i \in \sim B$, and thus $\tilde{\gamma}(\vee\, B_i) = \vee B_i$. From the first relation above, it follows for any $\psi \in \mathcal{J}\mathrm{d}(\mathcal{B})$, that

$$\psi\,(\vee\, B_i) = \psi\tilde{\gamma}\,(\vee\, B_i) = \tilde{\varphi}\,\tilde{\gamma}\,(\vee\, B_i)\,.$$

But $\tilde{\gamma}\,(\vee\, B_i) = \vee\, B_i$, so that

$$\tilde{\varphi}\,(\vee B_i) = \psi\,(\vee B_i) \in \mathcal{B}.$$

In the same way, we also obtain

$$\tilde{\gamma}\tilde{\varphi}\,(\wedge\, B_i) = \tilde{\gamma}\,(\wedge\, B_i) = \psi\,(\wedge\, B_i) \in \mathcal{B}.$$

In other words, $\mathcal{B}$ is a *complete lattice* with respect to the ordering on $\mathcal{B}$ induced by $\leq$, i.e. any family B_i in $\mathcal{B}$ has a smallest upper bound $\tilde{\varphi}\,(\vee\, B_i)\,\mathcal{B}$ and a greatest lower bound $\tilde{\gamma}\,(\wedge\, B_i) \in \mathcal{B}$.

Conversely, let us assume that $\mathcal{B}$ is a complete lattice. Thus, for any $A \in \mathcal{L}$, the family $\{B : B \in \mathcal{B}, B \geq A\}$ has in $\mathcal{B}$ a greatest lower bound, which is

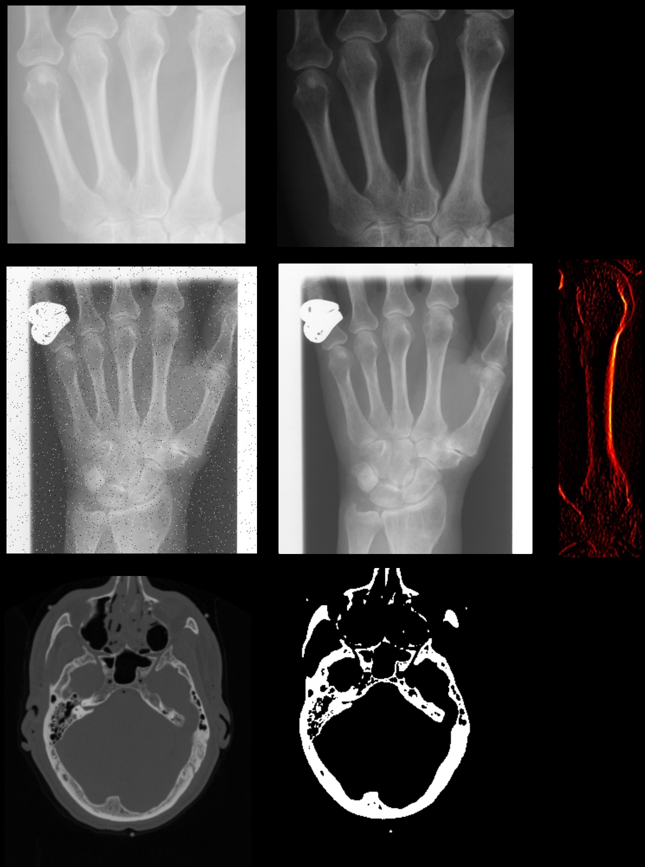
$$\tilde{\gamma}\,(\wedge\,\{B : B \in \mathcal{B}, B \geq A\}) = \tilde{\gamma}\,\tilde{\varphi}\,(A) \in \mathcal{B}.$$

But this implies $\mathcal{B}_{\psi_M} \subseteq \mathcal{B}$ for the filter $\psi_M = \tilde{\gamma}\,\tilde{\varphi}$. Conversely, for any

- Developed in 1964
- Theoretical work done in Paris
- Used for classification of minerals in cut stone
- Initially used for binary images

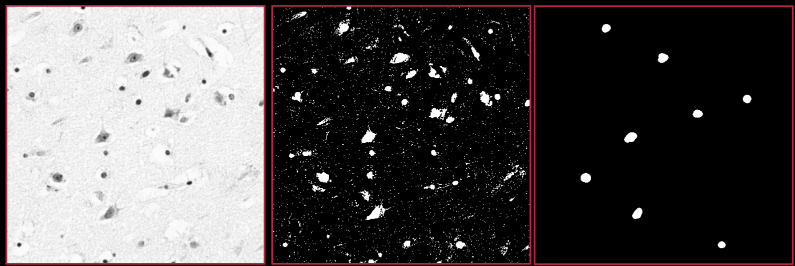
Do not worry! We use a much less theoretical approach!

Relevance?

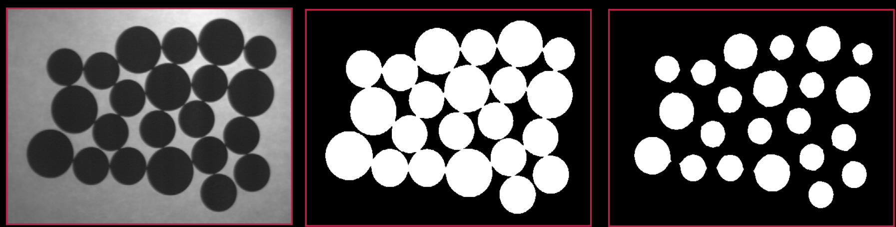


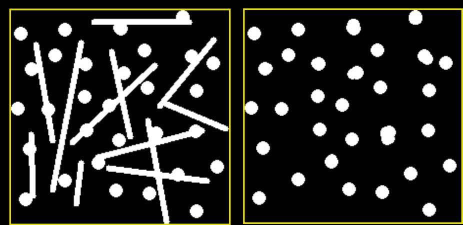
- Point wise operations
- Filtering
- Thresholding
  - Gives us objects that are separated by the background
- Morphology
  - Manipulate and enhance binary objects

What can it be used for?

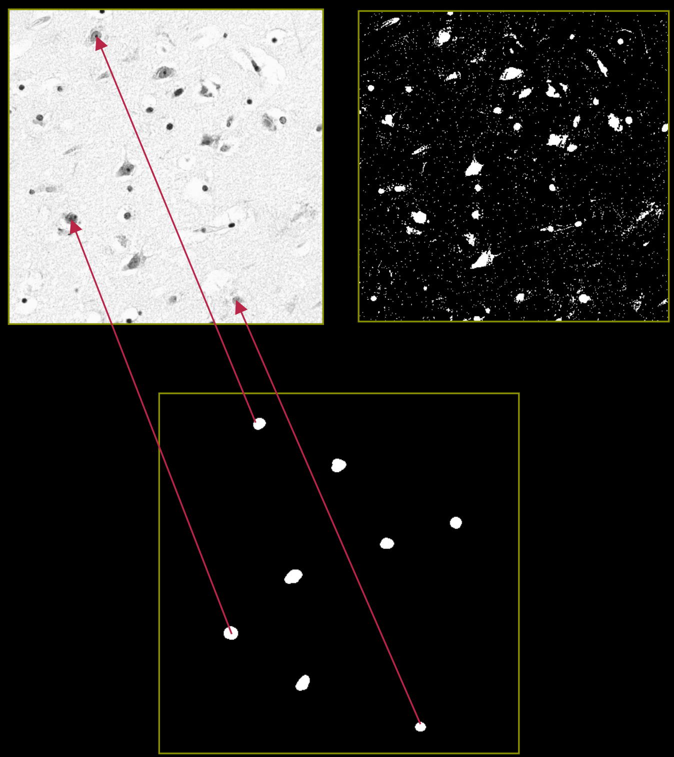


- Remove noise
  - Small objects
  - Fill holes
- Isolate objects
- Customized to specific shapes



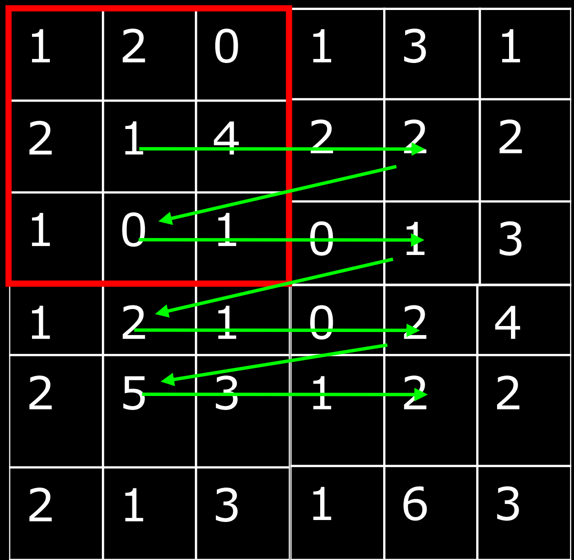


How does it work?



- Grayscale image
- Preprocessing
  - Inversion
- Threshold => Binary image
- Morphology

Filtering and morphology



- Filtering
  - Gray level images
  - Kernel
  - Moves it over the input image
  - Creates a new output image

Filtering and morphology

0	1	0
1	1	1
0	1	0

Disk

1	1	1
1	1	1
1	1	1

Box

- Filtering
  - Gray level images
  - Kernel
  - Moves it over the input image
  - Creates a new output image
- Morphology
  - Binary images
  - Structuring element (SE)
  - Moves the SE over the input image
  - Creates a new binary output image

1D Morphology

Input image

1	0	0	0	1	1	1	0	1	1

↓

Structuring Element (SE)

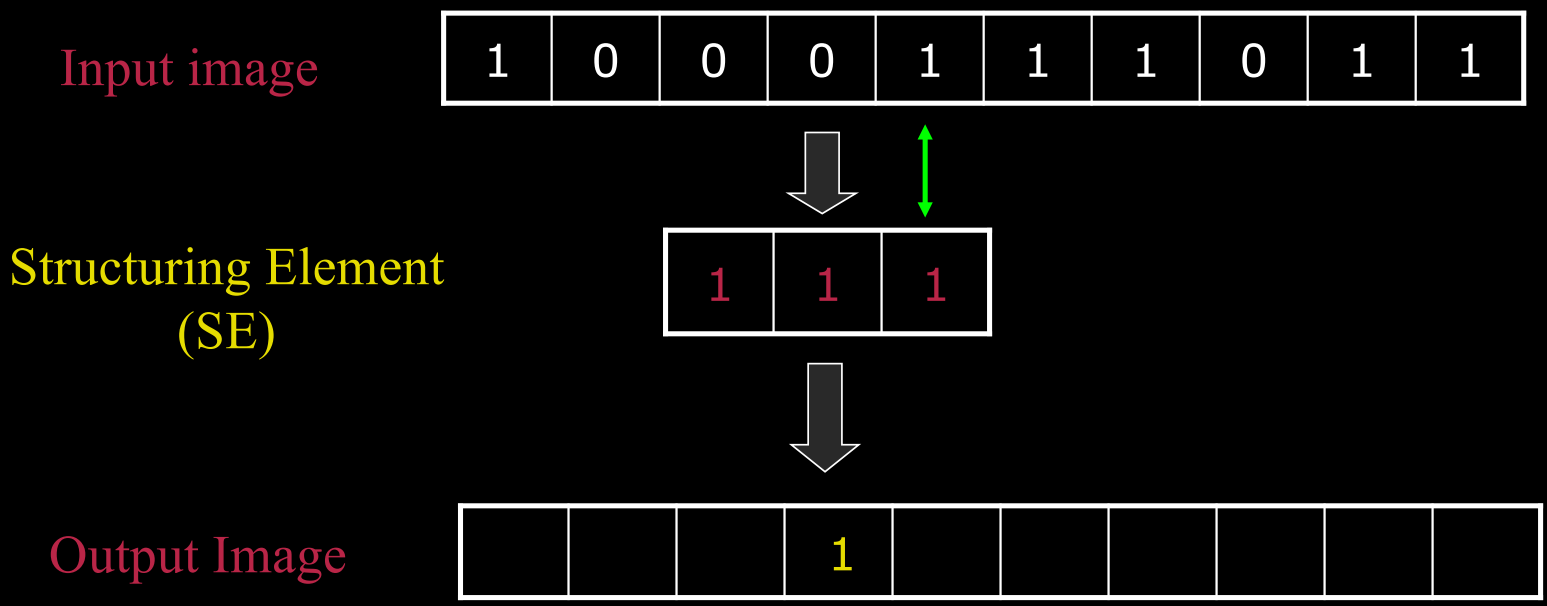
1	1	1

↓

Output Image

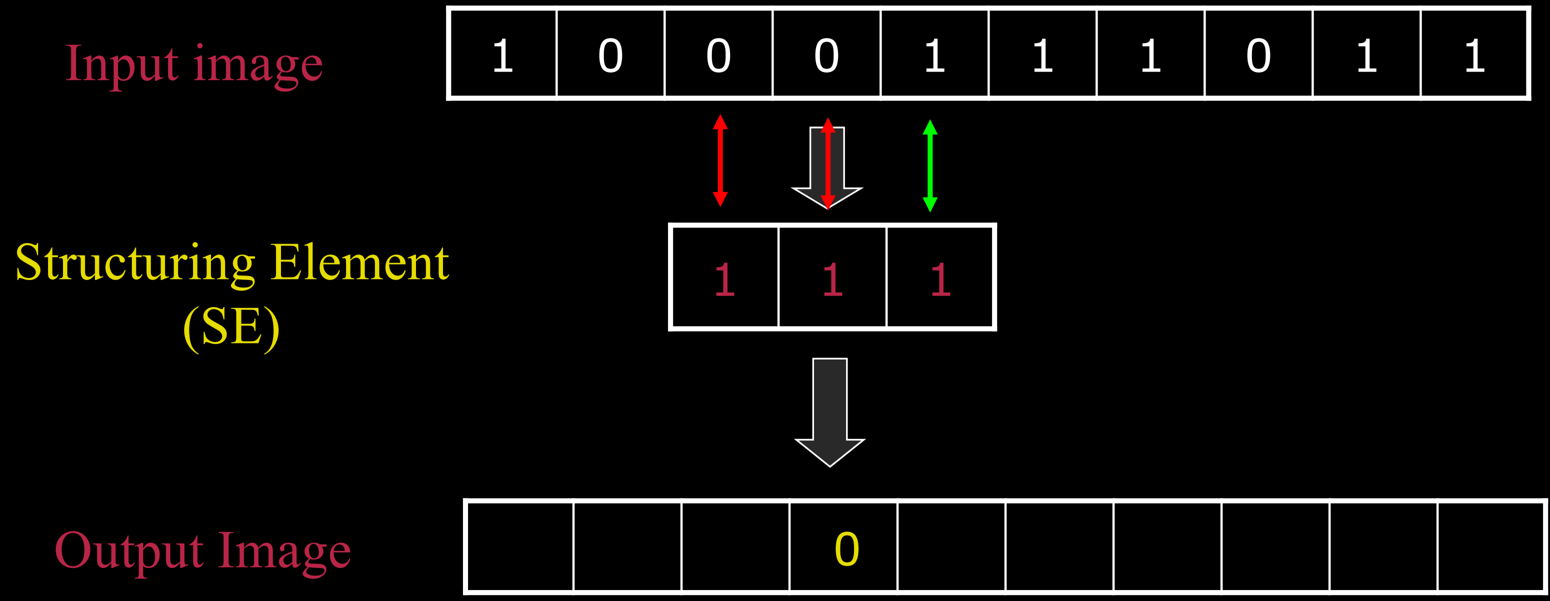
			?						

1D Morphology : The hit operation



- If just one 1 in the SE match with the input
  - output 1
- else
  - output 0

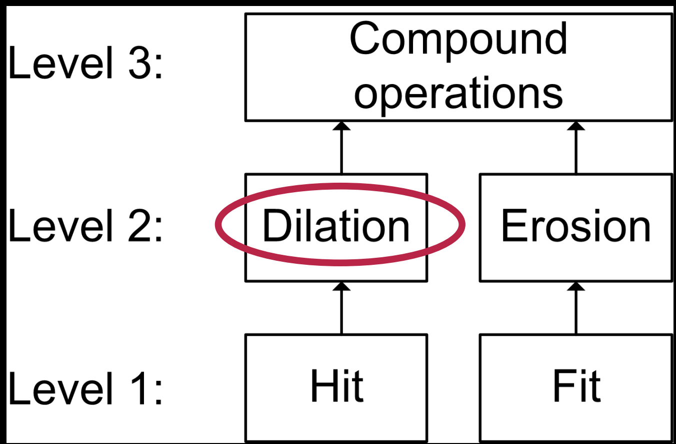
1D Morphology : The fit operation



- If all 1 in the SE match with the input
    - output 1
- else
    - output 0

1D Morphology : Dilation

- Dilate : To make wider or larger
  - Dansk : udvide
- Based on the *hit* operation



1D Dilation example

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

$$g(x) = f(x) \oplus SE$$

to make bigger

Output Image

	1								

Hit

- If just one 1 in the SE match with the input
    - output 1
- else
    - output 0

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0							

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1						

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1					

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1				

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1			

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1	1		

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

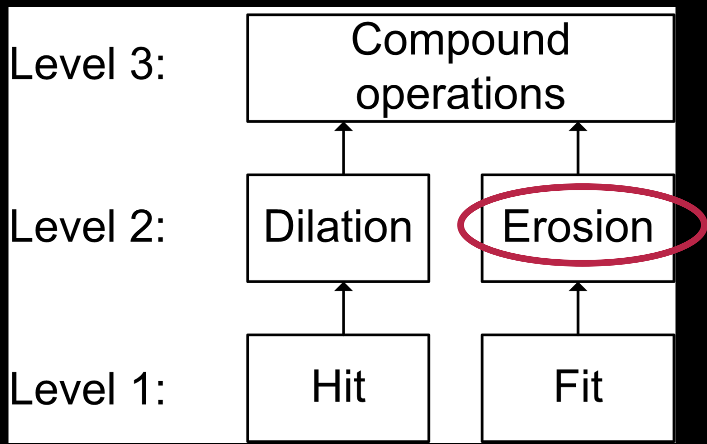
Output Image

	1	0	1	1	1	1	1	1	

The object gets bigger and holes are filled!

1D Morphology : Erosion

- Erode : To wear down (*Waves eroded the shore*)
  - Dansk : tære, gnave
- Based on the *fit* operation



Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

$$g(x) = f(x) \ominus SE$$

to make smaller

Output Image

	0								

Fit

- If all 1 in the SE match with the input
  - output 1
- else
  - output 0

Quiz 10: 1D Erosion

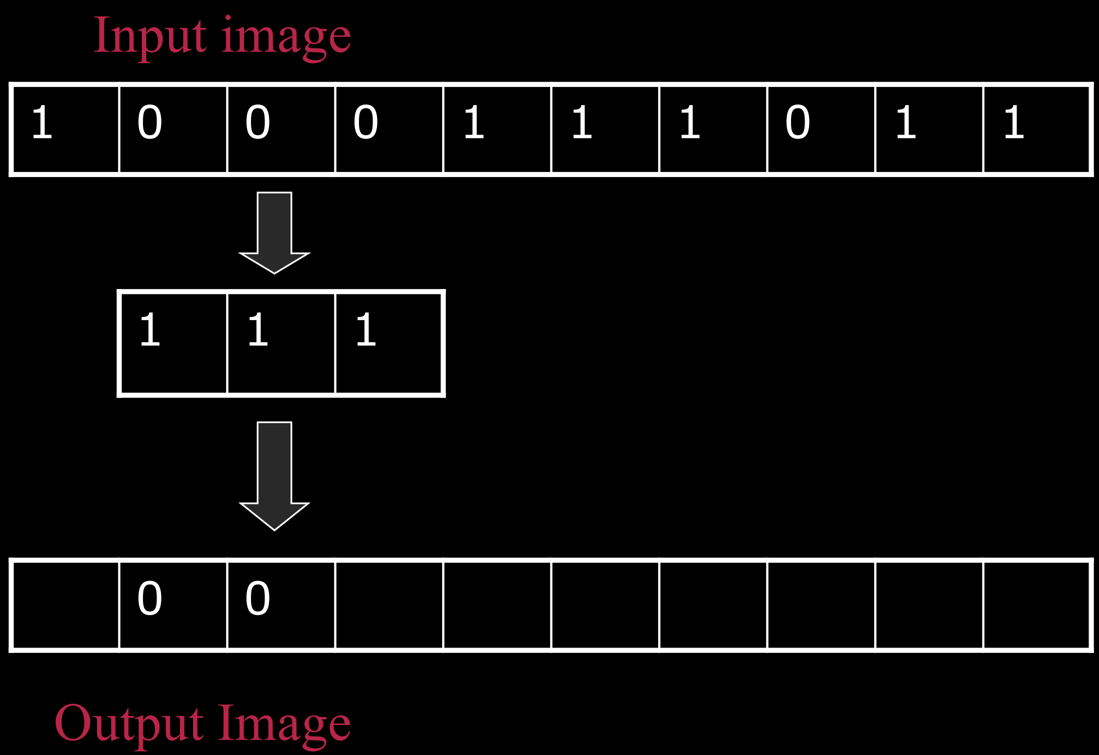
A) 0 1 0 0 1 1 0 0

B) 0 0 1 0 1 0 0 0

C) 0 0 0 0 1 0 0 0

D) 0 0 1 0 0 0 0 1

E) 0 1 0 0 0 1 0 0



Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0	1	0	0	0	

The object gets smaller

Structuring Element (Kernel)

0	1	0
1	1	1
0	1	0

Disk

1	1	1
1	1	1
1	1	1

Box

		1	1	1		
	1	1	1	1	1	
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
	1	1	1	1	1	
		1	1	1		

- Structuring Elements can have varying sizes
- Usually, element values are 0 or 1, but other values are possible (including none!)
- Structural Elements have an origin
- Empty spots in the Structuring Elements are *don't cares*!

Structuring Element Origin

0	1	0
1	1	1
0	1	0

1	1	1
1	1	1
1	1	1

- The origin is not always the center of the SE

Special structuring elements

- Structuring elements can be customized to a specific problem

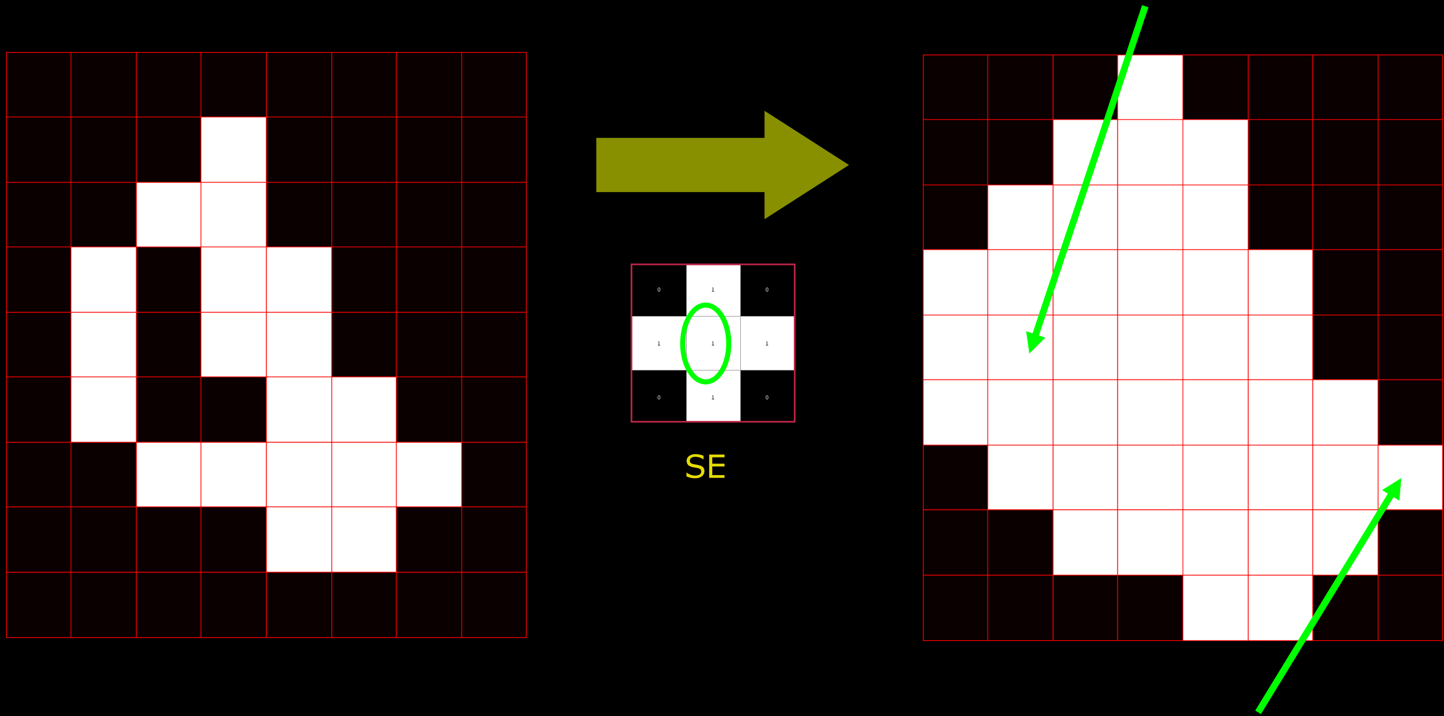
0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0

Diamond

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

Line

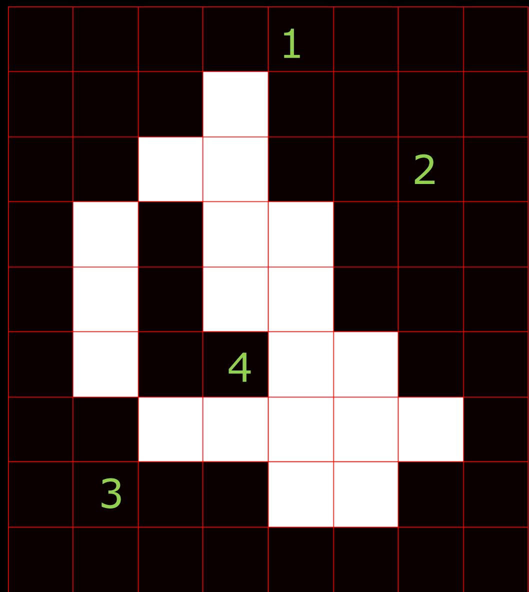
Dilation on images - disk

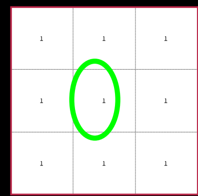


$$g(x,y) = f(x,y) \oplus SE$$

Quiz 11: Dilation on image - box

	1	2	3	4
A)	1	0	1	1
B)	0	1	0	0
C)	0	1	1	1
D)	0	1	0	1
E)	1	1	0	1

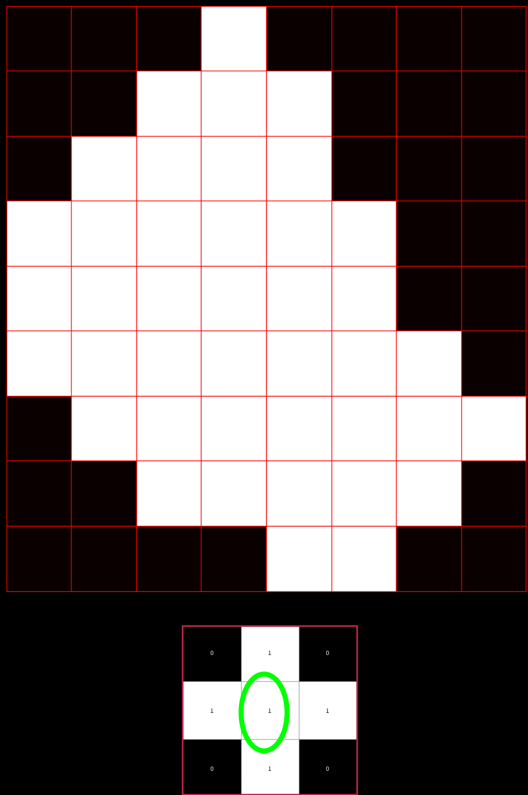


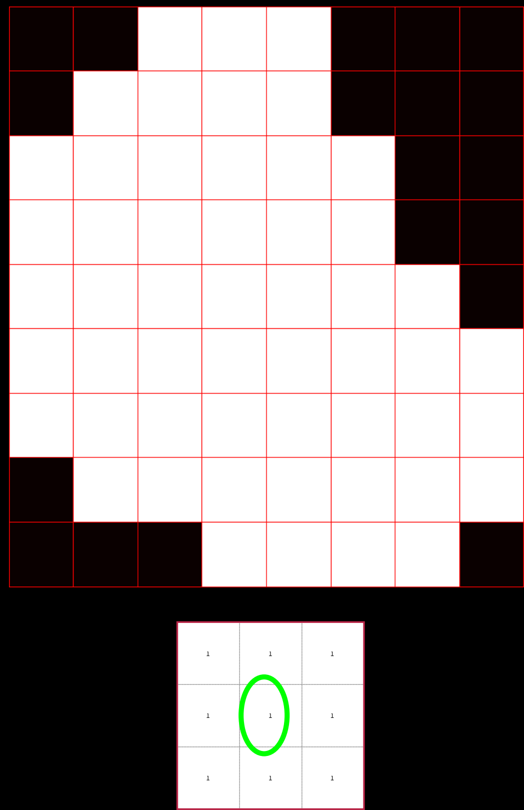


SE

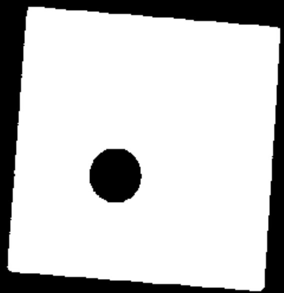
$$g(x,y) = f(x,y) \oplus SE$$

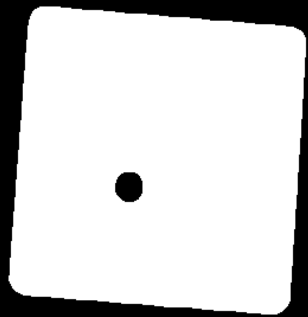
Dilation – the effect of the SE





Dilation Example





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

- Round structuring element (disk)
- Creates round corners

Quiz 12: Threshold and dilation

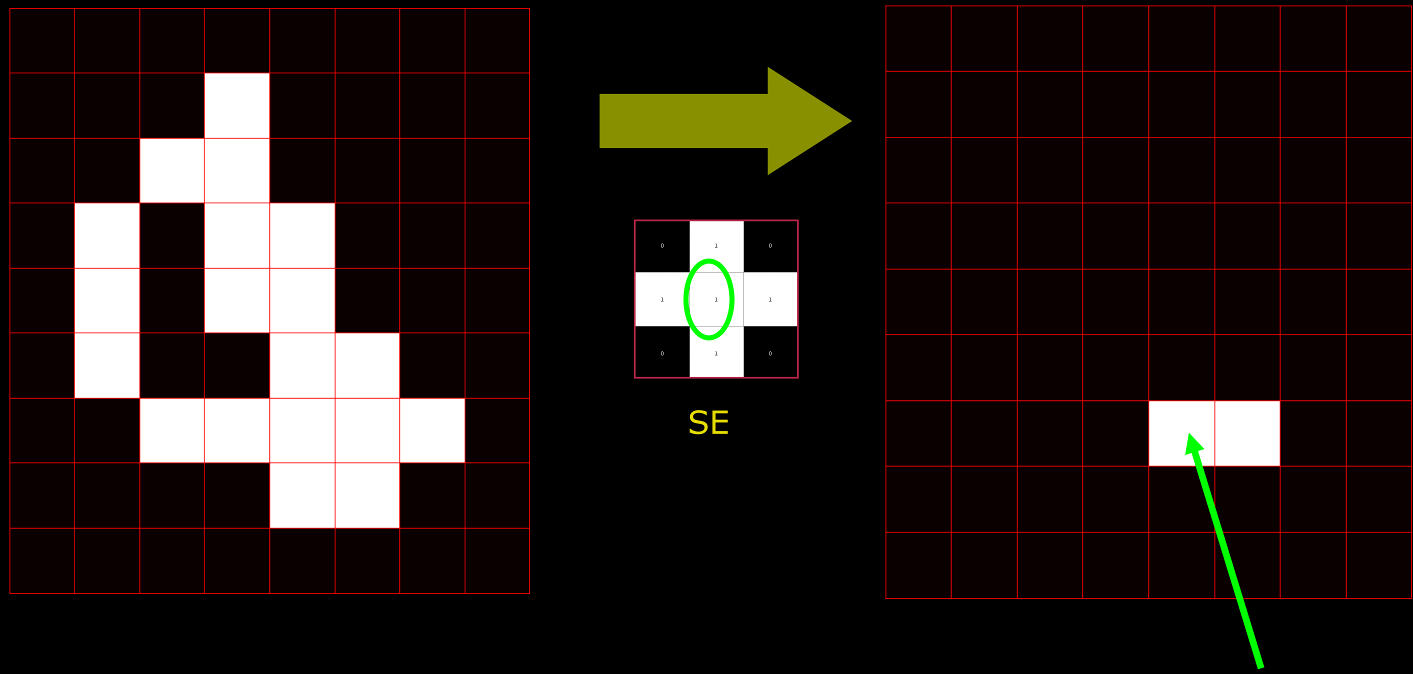
A threshold of 200 is applied to the image and the result is a binary image. Now a dilation is performed with the structuring element below. How many foreground pixels are there in the resulting image?

145	56	86	42	191
19	33	41	255	115
14	240	203	234	21
135	120	209	167	58
199	3	135	176	116

- A) 14
- B) 17
- C) 6
- D) 3
- E) 12

Answer:

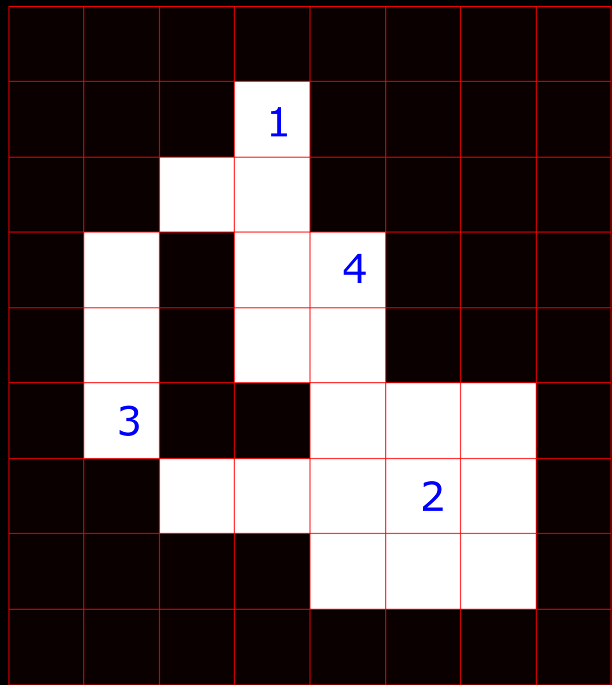
Erosion on images - disk

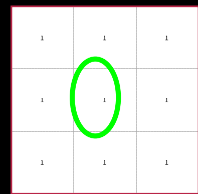


$$g(x, y) = f(x, y) \ominus SE$$

Quiz 13: Erosion on images - box

	1	2	3	4
A)	0	0	1	0
B)	1	0	1	0
C)	0	1	1	0
D)	0	1	0	0
E)	1	0	0	0

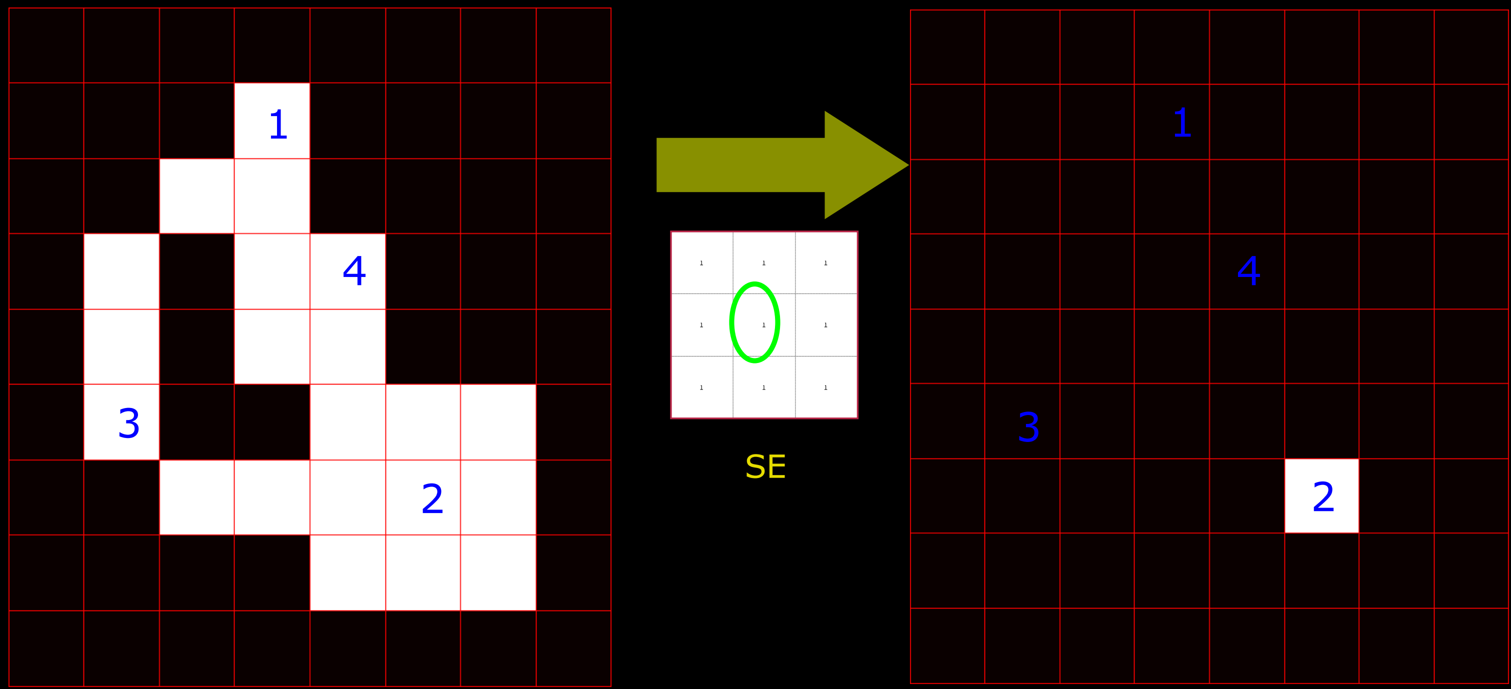




SE

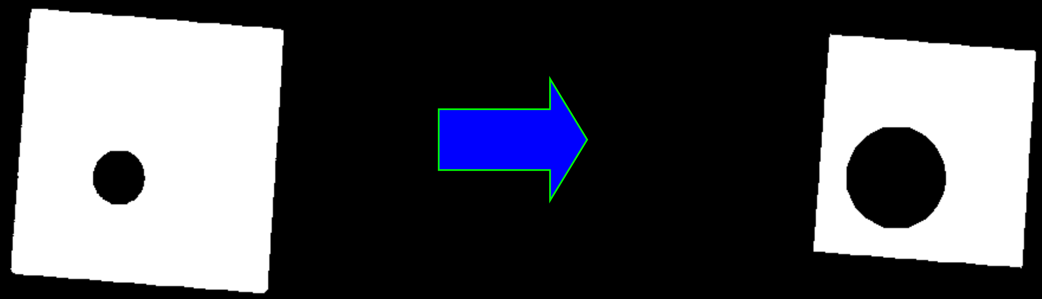
$$g(x,y) = f(x,y) \ominus SE$$

Erosion on images – box (square)



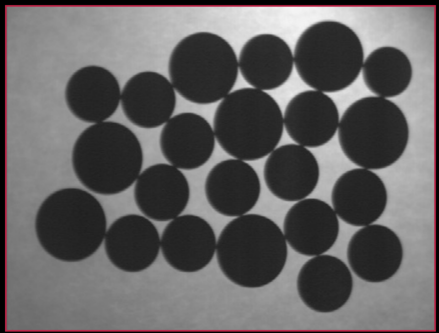
$$g(x,y) = f(x,y) \ominus SE$$

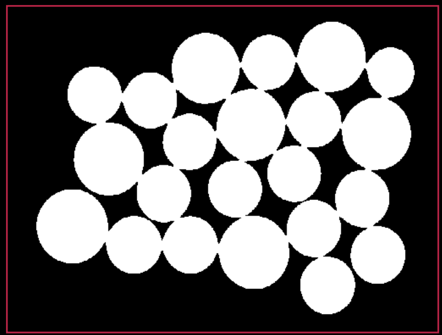
Erosion example

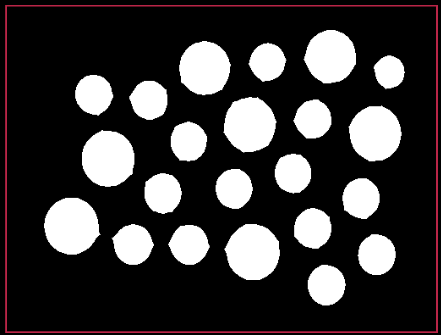


Counting Coins

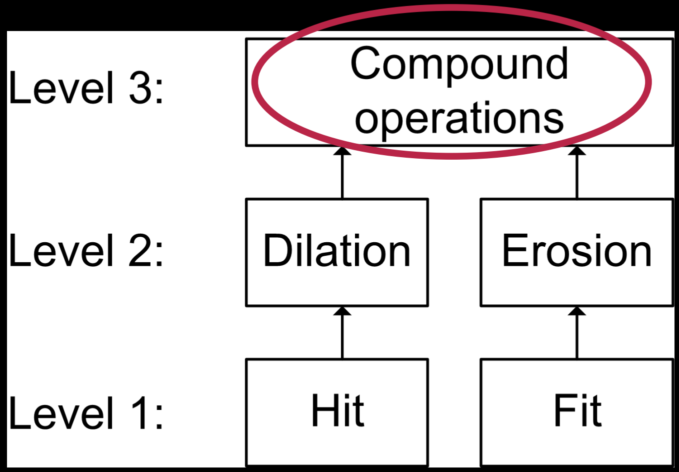
- Counting these coins is difficult because they touch each other!
- Solution: Threshold and Erosion separates them!
- More on counting next time!







Compound operations

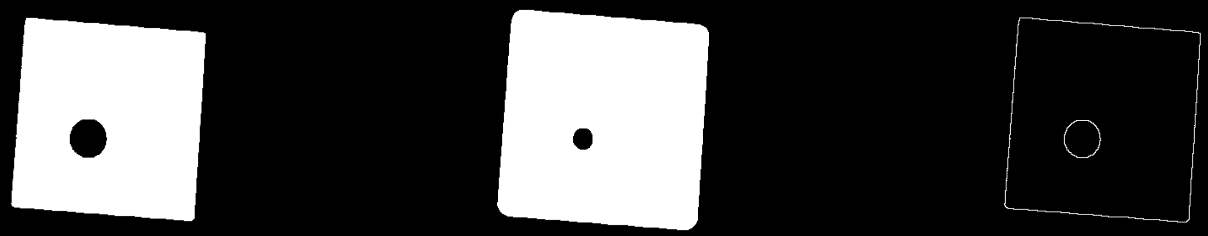


- Compound
  - *made of two or more separate parts or elements*
- Combining Erosion and Dilation into more advanced operations
  - Finding the outline
  - Opening
    - Isolate objects and remove small objects (better than Erosion)
  - Closing
    - Fill holes (better than Dilation)

Finding the outline

1. Dilate input image (object gets bigger)
2. Subtract input image from dilated image
3. The outline remains!

$$g(x,y) = (f(x,y) \oplus SE) - f(x,y)$$



Opening

- Motivation: Remove small objects BUT keep original size (and shape)
- Opening = Erosion + Dilation
  - Use the same structuring element!
  - Similar to erosion but less destructive
- Math:

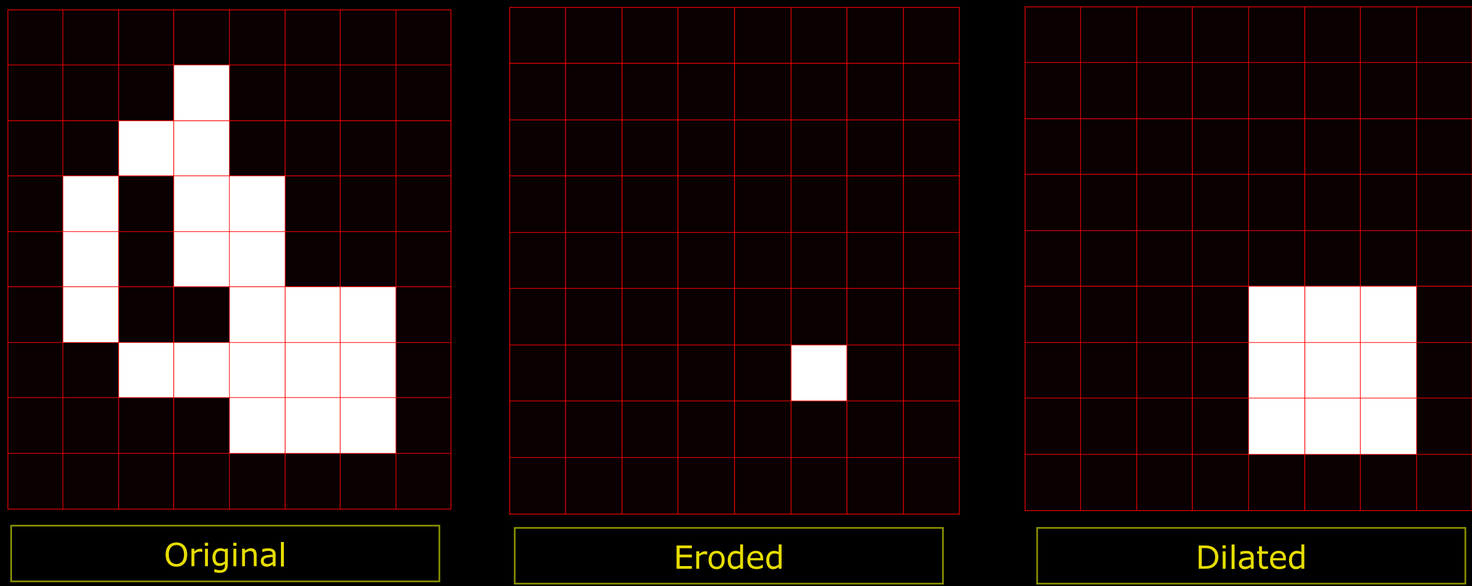
$$g(x,y) = f(x,y) \circ SE = (f(x,y) \ominus SE) \oplus SE$$

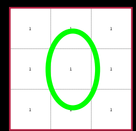
- Opening is **idempotent:** Repeated operations have no further effects!

$$f(x,y) \circ SE = (f(x,y) \circ SE) \circ SE$$

Opening

$$g(x,y) = \left(f(x,y) \ominus SE\right) \oplus SE$$



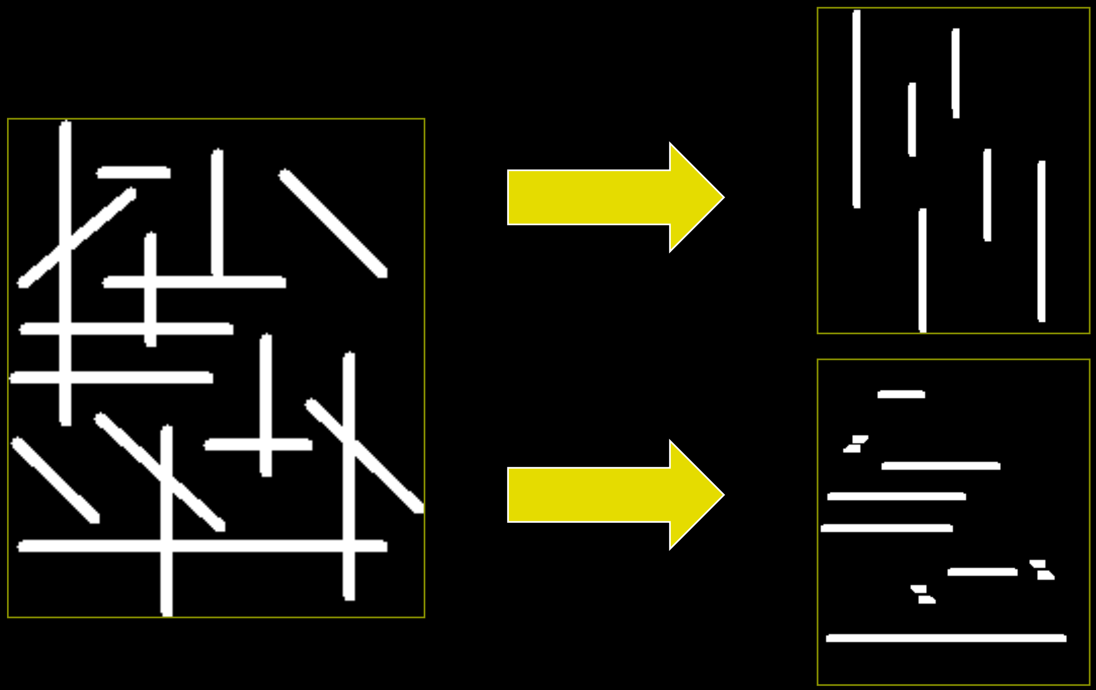


SE

Opening = erosion+dilation

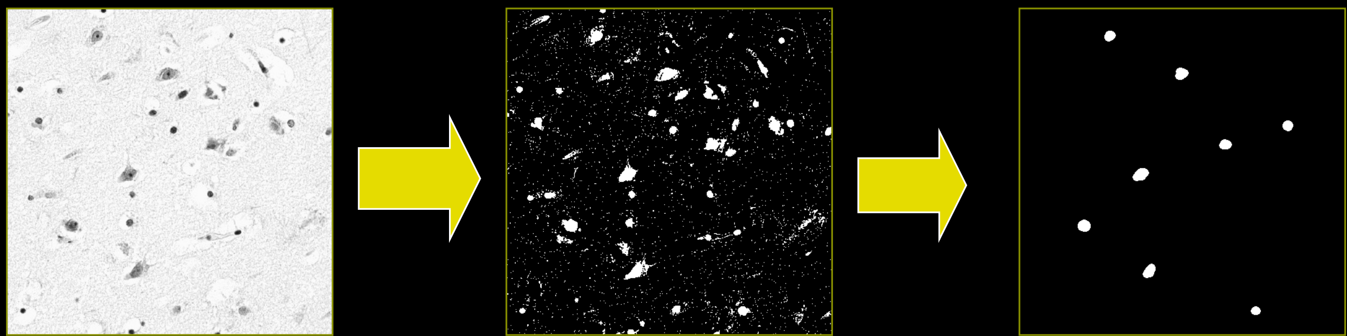
Opening Example

- 9x3 and 3x9 Structuring Elements



Opening example

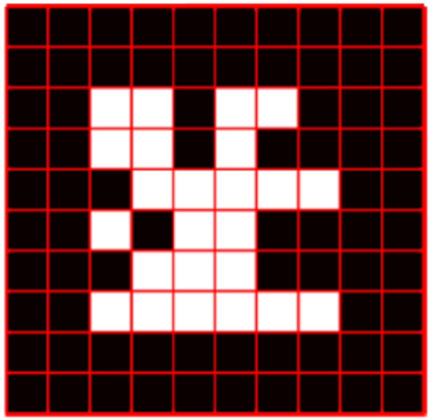
- Size of structuring element should fit into the smallest object to keep
- Structuring Element: 11 pixel disc

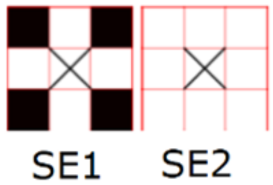


Quiz 14: Compound operations on image

The compound morphological operation seen below is applied to the image. How many foreground pixels are there in the resulting image?

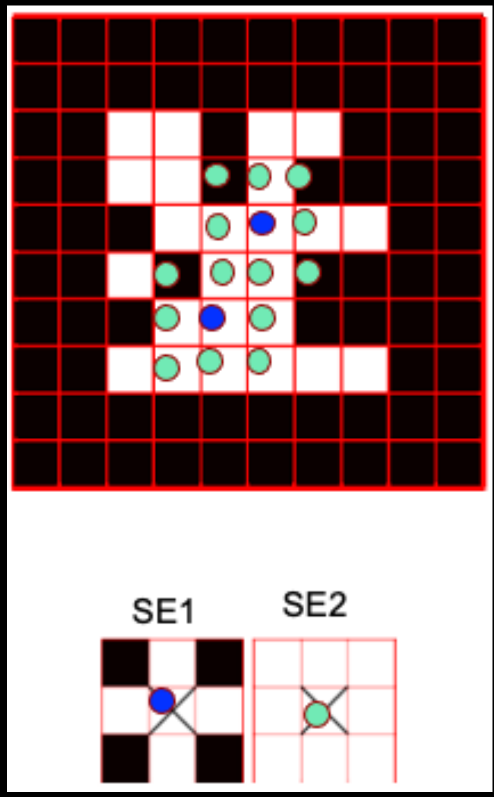
$$(\mathrm{I} \ominus \mathrm{SE1}) \oplus \mathrm{SE2},$$





A) 3
B) 23
C) 11
D) 36
E) 16

Answer:



Closing

- Motivation: Fill holes BUT keep original size (and shape)
- Closing = Dilation + Erosion
  - Use the same structuring element!
  - Similar to Dilation but less destructive
- Math:

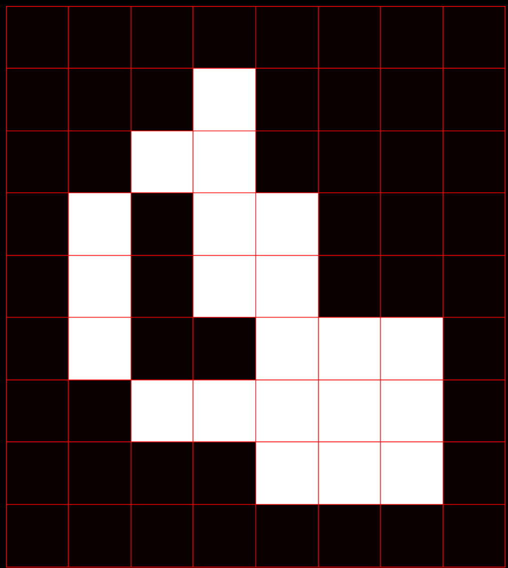
$$g(x,y) = f(x,y) \bullet SE = (f(x,y) \oplus SE) \ominus SE$$

- Closing is **idempotent:** Repeated operations have no further effects!

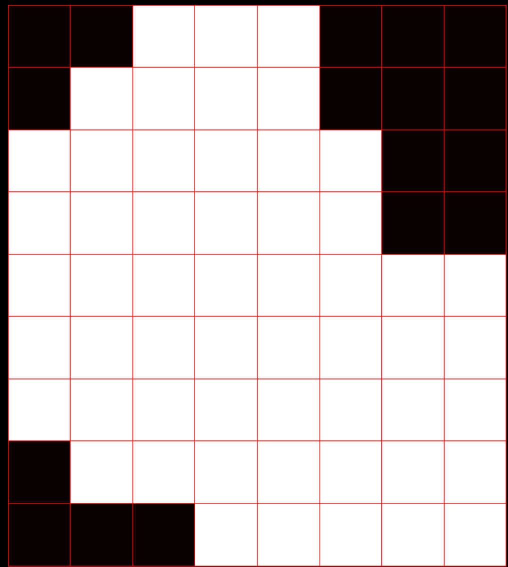
$$f(x,y) \circ SE = (f(x,y) \circ SE) \circ SE$$

Closing

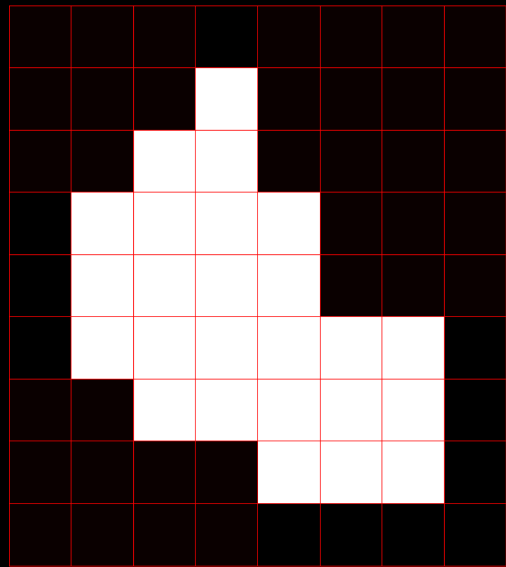
$$g(x,y) = (f(x,y) \oplus SE) \ominus SE$$



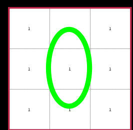
Original



Dilated



Eroded

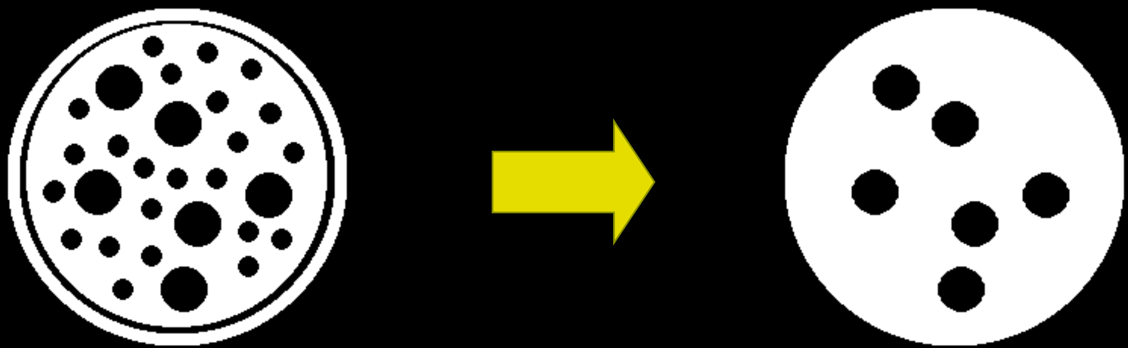


SE

Closing = dilation + erosion

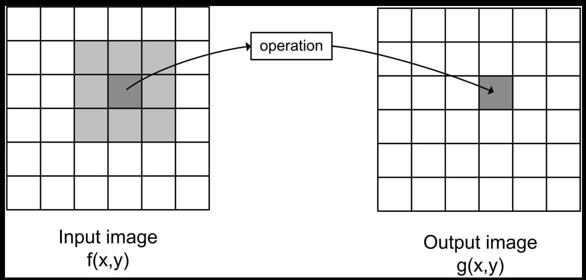
Closing Example

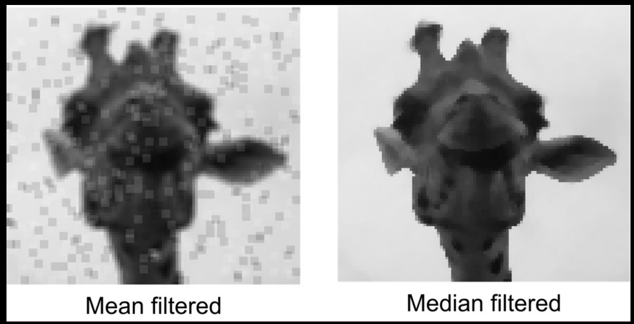
- Closing operation with a 22 pixel disc
- Closes small holes

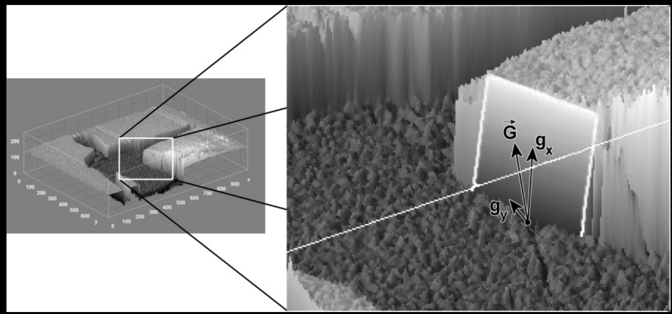


What did we learn today

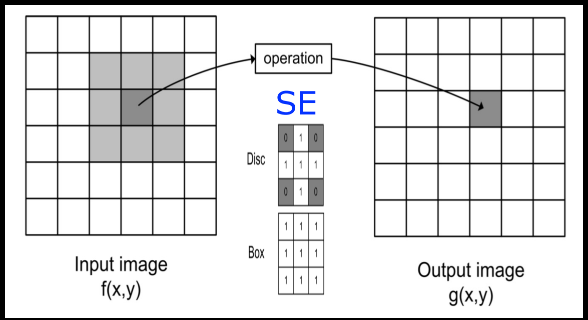
Neighbourhood Processing

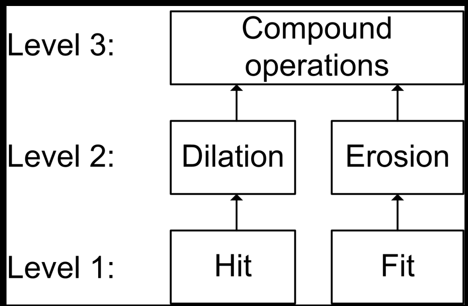


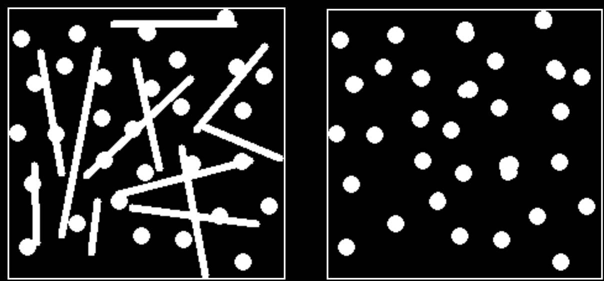




Morphology of binary images







Next week:
Blob Analysis and object classification