Medical Image Analysis, 2018
Introduction
KALLE ÅSTRÖM
Medical Image Analysis
Introduction

• Who are we (the lecturers)
• Medical Image Analysis at Lund University
• Information about the course
  – Learning objectives
  – Competences and skills -> Medical Image Analysis Problems
  – Course content
• Teaching
• Course Material
• Examples of medical imaging
Who are we

• Einar Heiberg
  – Associate professor, Numerical Analysis, Clinical Physiology, Medical Image Analysis

• Anders Heyden
  – Professor, Mathematics, Image Analysis

• Niels Christian Overgaard
  – Associate professor, Mathematics, Image Analysis

• Kalle Åström
  – Professor, Mathematics, Image Analysis
Medical Image Analysis at Lund University

- Centre for mathematical Sciences
  - http://www.maths.lu.se/forskning/forskargrupper/
  - Mathematical imaging group
  - http://www.maths.lu.se/forskning/forskargrupper/mathematical-imaging-group/Signal processing

- Dept of Clinical Physiology - Cardiac MR Group
  - http://www.med.lu.se/klinvetlund/klinisk_fysiologi/forskning/cardiac_mr_group

- Dept of Biomedical Engineering
  - http://bme.lth.se/research/

- Lund University Bioimaging Center
  - https://www.med.lu.se/bioimaging_center

- LABIB
  - https://www.lth.se/labib/
Learning objectives

• Describe different image acquisition techniques used in medical imaging, e.g. Röntgen, CT, MR, ultrasound, PET, Scint and SPECT.

• Explain and use medical image analysis algorithms to perform registration, segmentation and classification

• Decide on appropriate algorithms for solving medical image analysis problems

• Implement automated medical analysis systems

• Validate the results of automated medical analysis systems
Competences and skills

For a passing grade the student must

• in an engineering manner be able to use computer packages to solve problems in medical image analysis.

• be able to independently apply basic methods in medical image processing to problems which are relevant in medical applications or research.

• with proper terminology, in a well structured way and with clear logic be able to explain the solution to a problem in medical image analysis.
Detection and Diagnosis of Kidney Lesions
Scandinavian Conference on Image Analysis, 2011

<table>
<thead>
<tr>
<th></th>
<th>LDA</th>
<th>QDA</th>
<th>ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area under ROC curve (AUC)</td>
<td>0.964</td>
<td>0.935</td>
<td>0.960</td>
</tr>
<tr>
<td>Sensitivity (%)</td>
<td>96.5</td>
<td>96.5</td>
<td>96.5</td>
</tr>
<tr>
<td>Specificity (%)</td>
<td>84.8</td>
<td>61.2</td>
<td>83.4</td>
</tr>
<tr>
<td>Positive Predictive value (%)</td>
<td>35.0</td>
<td>17.4</td>
<td>32.9</td>
</tr>
<tr>
<td>Negative Predictive value (%)</td>
<td>99.7</td>
<td>99.5</td>
<td>99.6</td>
</tr>
<tr>
<td>Mis-classification rate (%)</td>
<td>14.2</td>
<td>36.0</td>
<td>15.6</td>
</tr>
</tbody>
</table>
Generate a lot of features using standard image processing techniques.

Let machine learning algorithm handle the potential over fitting.

We use random forest which is fast to train and extremely fast to classify, 50μs.

High accuracy and confidence on each classification.

The confidence measure can be used in semi-supervised setup.

Accuracy as a function of reject rate. We can remove the result we are most uncertain of. 11.1% removed give perfect result.
Diagnosis of Pulmonary Embolism
European Journal of Nuclear Medicine, 2000

The area is 85.5706% of the total area.
Segmentation – shape variation methods
Understanding both appearance and shape
SCINT - Heart
Segmentation results
Exini Diagnostics
MR – knee injuries
SPECT – brain (dementia)
# Gated SCINT - heart

<table>
<thead>
<tr>
<th></th>
<th>EDV</th>
<th>ESV</th>
<th>EDV</th>
<th>ESV</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAFU</td>
<td>46 ml</td>
<td>21 ml</td>
<td>185 ml</td>
<td>71 ml</td>
</tr>
<tr>
<td>QGS</td>
<td>45 ml</td>
<td>21 ml</td>
<td>201 ml</td>
<td>122 ml</td>
</tr>
<tr>
<td>TRUE</td>
<td>57 ml</td>
<td>22 ml</td>
<td>186 ml</td>
<td>72 ml</td>
</tr>
</tbody>
</table>
Automated Pharmacokinetic Analysis
Scandinavian Conference on Image Analysis, 2011
Longitudinal expansion of left ventricle
Hearing aid manufacturing
Automated brain segmentation
Automated brain segmentation
Dimensionality

- 1D
- 3D
- 2D
- 3D+T
- 2D+T
- 3D+T+T
Dimensionality

- Outer dimensions
- Inner dimensions

- Scalar: Pixel intensity, attenuation, temperature, radiation count, energy, ...

- Vector: Velocity, spectrum, ...

- Tensor: Strain, strain-rate, stress, ....
Colour vs greyscale

- We can perceive about 100 shades of grey
- We can perceive > 100 000 colours
Contrast

Original

“Contrast enhanced”
Transfer curve

Input intensity

Original

Result
Transfer curve

Input intensity

Original

Result
Contrast agents

- Iron particles (MRI)
- Gadolinium (MRI)
- Radio nuclides (SPECT, PET)
- Bubbles (ultrasound)
- Staining (histology)

- Nanoparticles with multiple contrast agents
Resolution
What is noise?
What is noise?
Signal to Noise Ratio (SNR)

• $\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}}$

• $\text{SNR} = \frac{\mu}{\sigma}$
Accelerated imaging

SNR not constant over the image!
Course content
Topics

• Introduction, validation, databases, dicom
• Ethics, regulatory aspects
• Image registration
• Image segmentation
• Machine learning

• Examples
• Invited talks
Teaching form

- Lectures (16 lectures)
- 4 assignments
  - OK on assignments + short oral exam -> pass
  - Dates, times and room for supervision on assignments will be posted on the homepage
  - First assignment handed out on friday 9/11

Course requirements/Exam:
In the course, there are 4 mandatory hand-in exercises and a short oral exam:

<table>
<thead>
<tr>
<th>nr</th>
<th>Handed out</th>
<th>title</th>
<th>deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9/11</td>
<td>Registration</td>
<td>25/11</td>
</tr>
<tr>
<td>2</td>
<td>12/11</td>
<td>Shape Statistics</td>
<td>2/12</td>
</tr>
<tr>
<td>3</td>
<td>26/11</td>
<td>Machine Learning</td>
<td>16/12</td>
</tr>
<tr>
<td>4</td>
<td>4/12</td>
<td>Segmentation</td>
<td>8/1</td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Room</td>
<td>Topic</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>--------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Mon 5/11</td>
<td>8-10</td>
<td>E:1406</td>
<td>Introduction, Basic Concepts</td>
</tr>
<tr>
<td>Fri 9/11</td>
<td>13-15</td>
<td>MH:G</td>
<td>Registration 1</td>
</tr>
<tr>
<td>Mon 12/11</td>
<td>8-10</td>
<td>E:1406</td>
<td>Segmentation 1</td>
</tr>
<tr>
<td>Tue 13/11</td>
<td>13-15</td>
<td>MA:3</td>
<td>Registration 2</td>
</tr>
<tr>
<td>Fri 16/11</td>
<td>13-15</td>
<td>MH:G</td>
<td>Registration 3</td>
</tr>
<tr>
<td>Mon 19/11</td>
<td>8-10</td>
<td>E:1406</td>
<td>Segmentation 2</td>
</tr>
<tr>
<td>Fri 23/11</td>
<td>13-15</td>
<td>MH:G</td>
<td>Segmentation 3</td>
</tr>
<tr>
<td>Mon 26/11</td>
<td>8-10</td>
<td>E:1406</td>
<td>Machine Learning 1</td>
</tr>
<tr>
<td>Fri 30/11</td>
<td>13-15</td>
<td>MH:G</td>
<td>Machine Learning 2</td>
</tr>
<tr>
<td>Mon 3/12</td>
<td>8-10</td>
<td>E:1406</td>
<td>Machine Learning 3</td>
</tr>
<tr>
<td>Tue 4/12</td>
<td>13-15</td>
<td>MA:3</td>
<td>Dicom, Regulatory issues</td>
</tr>
<tr>
<td>Fri 7/12</td>
<td>13-15</td>
<td>MH:G</td>
<td>Fast Marching</td>
</tr>
<tr>
<td>Mon 10/12</td>
<td>8-10</td>
<td>E:1406</td>
<td>Machine Learning 4</td>
</tr>
<tr>
<td>Fri 14/12</td>
<td>13-15</td>
<td>MH:G</td>
<td>Ethics</td>
</tr>
<tr>
<td>Mon 17/12</td>
<td>8-10</td>
<td>E:1406</td>
<td>Extra</td>
</tr>
<tr>
<td>Fri 21/12</td>
<td>13-15</td>
<td>MH:G</td>
<td>Review, Wrap-up</td>
</tr>
</tbody>
</table>
Course material

- Website, homepage
- Lecture notes, papers
  - On website:
- Assignments
- Schedule
Masters thesis suggestion of the day

- Detection and grading of cancer in histopathological images
- Pathological samples
- Staining
- Scanning
- High resolution
- GB sized images
- Large field of view
The best prognostic and predictive biomarker in PCA
... but it is subjective and with low reproducibility
... attempts to standardize ...
A. Classification baseline using Deep Learning

- Manual microscopic inspection of prostate biopsies
- Stained with haematoxylin and eosin (H&E)
- The Gleason grade determines the diagnosis
- Classify each cutout into one of the following four classes:
D. Cycle-GAN

- Two sets of images (X and Y)
- Train
  - Converter X→Y(X)
  - Converter Y→X(Y)
  - Discriminator X vs X(Y)
  - Discriminator Y vs Y(X)
- Loss
  \[ X = X(Y(X)) \]
  \[ Y = Y(X(Y)) \]

D. Cycle-GAN

- Two sets of images (X and Y)
- Train
  - Converter X→Y(X)
  - Converter Y→X(Y)
  - Discriminator X vs X(Y)
  - Discriminator Y vs Y(X)
- Loss
  - $X = X(Y(X))$
  - $Y = Y(X(Y))$

D. Cycle-GAN – promising and interesting but might introduce features if not handled correctly
Review of some topics in image analysis
After the course

• You should be able to develop and test your own image analysis system
• You should have tools for understanding and working with big data
• You should have improved your skills in programming and modelling.

• Math, statistics, programming, image analysis
Testing your system

- Image analysis systems
  - Often complex and varying data
  - Often a system of systems
- Important to test your system
- Questions
  - Obtain data
  - Obtain 'ground truth' ('Gold Standard')
  - Construct benchmark scripts
  - Visualize the results
- Adress these questions early in a project
Images are functions

- Images are functions. Each pixel measures brightness

\[ f : \mathbb{Z}^2 \rightarrow \mathbb{Z} \]

Source: S. Narasimhan
Continuous Model

An image can be seen as a function

$$f : \Omega \mapsto \mathbb{R}_+,$$

where $\Omega = \{ (x, y) \mid a \leq x \leq b, c \leq y \leq d \} \subseteq \mathbb{R}^2$ and $\mathbb{R}_+ = \{ x \in \mathbb{R} \mid x \geq 0 \}$. $f(x, y)$ = intensity at point $(x, y)$ = gray-level

($f$ does not have to be continuous)

$0 \leq L_{\text{min}} \leq f \leq L_{\text{max}} \leq \infty$

$[L_{\text{min}}, L_{\text{max}}] = \text{gray-scale}$
Discrete Image Model

- Discretize $x, y \rightarrow$ sampling $M$ rows, $N$ columns
- Discretize $f \rightarrow$ quantization
  - (often in $2^m$ levels)
  - Color depth
    - "8 bit grayscale", $2^8 = 256$ levels, 0-255

- Decrease $f : \Omega \rightarrow \mathbb{Z}$
  - Chess patterns
- Decreasing $m$
  - False contours

\[ \Omega \subset \mathbb{Z}^2 \]
Sampling

Quantization
Interpolation

- Discrete image \( f : \mathbb{Z}^2 \rightarrow \mathbb{R} \)
- Continuous image \( F : \mathbb{R}^2 \rightarrow \mathbb{R} \)
- Going from \( F \) to \( f \) (sampling)
  \[ f(i, j) = D(F)(i, j) = F(i, j) \]
- Going from \( f \) to \( F \) (interpolation)
  \[ F_h(x, y) = I_h(f)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(x - i, y - j) f(i, j) \]

Different choices of \( h \) (different humps)
-> different types of interpolation
Interpolation

• Discrete image $f$ 
  \[ f : \mathbb{Z}^2 \rightarrow \mathbb{R} \]
• Continuous image $F$ 
  \[ F : \mathbb{R}^2 \rightarrow \mathbb{R} \]
• If the function $F$ is square integrable, i.e.
  \[ \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} |F(x, y)|^2 dx dy \leq \infty \]
• Is bounded.
• If also the fourier transform is zero outside
• Then 
  \[ [\pi, \pi] \times [-\pi, \pi]. \]

• Claude Shannon, 1949, Harry Nyquist, 1928
  \[ I(D(F)) = F. \]
Digital Geometry (bwlabel)
Finding connected components

>> bild = [1 1 0 0 0;1 0 0 0 1;0 0 0 0 1;1 1 0 1 1]

bild =

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

>> segmentering = bwlabel(bild)

segmentering =

1 1 0 0 0 0
1 0 0 0 0 3
0 0 0 0 3
2 2 0 3 3
Gray level transformations
Pixelwise operations

A simple method for image enhancement

**Definition**
Let \( f(x, y) \) be the intensity function of an image. A **gray-level transformation**, \( T \), is a function (of one variable)

\[
g(x, y) = T(f(x, y))
\]

\[
s = T(r)
\]

that changes from gray-level \( f \) to gray-level \( g \). \( T \) usually fulfills

- \( T(r) \) increasing in \( L_{min} \leq r \leq L_{max} \),
- \( 0 \leq T(r) \leq L \).

In many examples we assume that \( L_{min} = 0 \) och \( L_{max} = L = 1 \). The requirements on \( T \) being increasing can be relaxed, e.g. with inversion.
Histograms

The original gray-scales gives this image and this cumulative histogram.

This gray-scale transformation gives this image and this cumulative histogram.
• Machine Learning typically has two phases
  • Phase 1 – Training
    – A training dataset is used to estimate model parameters. Store these parameters. Code usually assumes that input are vectors
  • Phase 2 – Prediction
    – Once the parameters have been estimated, we can use the model to classify future data
All of these classification problems have in common:

- data - \( x \) (after segmentation, extract features)
- A number of classes

One would like to determine a class for every possible feature vector. Here we will assume that the features are represented as a column vector, i.e. \( x \in \mathbb{R}^n \),

\[
x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}
\]

One would like to compare the feature vector \( x \) with those that one usually gets with a number of classes. Let \( y \) denote the class index, i.e. the classes are \( y \in \omega_y = \{1, \ldots, M\} \) where \( M \) denotes the number of classes.

**Typical system:** Image - filtering - segmentation - features - classification
Assume that one feature vector $\mathbf{x}$ and class $y$ are drawn from a joint probability distribution. If one can calculate the probability that the class is $y = j$ given the measurements $\mathbf{x}$, i.e. the so-called posterior probability.

$$P(y = j | \mathbf{x})$$

The maximum a posteriori classifier is obtained as selecting the class $j$ that maximizes the posterior probability, i.e.

$$j = \underset{k}{\text{argmax}} P(y = k | \mathbf{x}).$$

It is often easier to model and estimate the likelihood $P(\mathbf{x} | y = j)$ and to model the prior $p(y = j)$. The a posteriori probabilities can then be calculated using the Bayes rule,

$$p(y = j | \mathbf{x}) = \frac{p(\mathbf{x} | y = j)p(y = j)}{p(\mathbf{x})}.$$
False Positives, False Negatives
ROC - Curve

- For two class problems - Negatives and Positives
- Negatives that are classified as negatives – True Negatives (TN)
- Positives that are classified as positives – True Positives (TP)
- Negatives that are classified as positives – False Positives (FP)
- Positives that are classified as negatives – False Negatives (FN)

- False Positive Rate TPR = FP/(FP+TN) -> x-axis
- True Positive Rate TPR = TP/(TP+FN) -> y-axis
K-means

1. Initialize cluster centers: \( c^0 \); \( t=0 \)

2. Assign each point to the closest center

\[
\delta^t = \arg\min_\delta \frac{1}{N} \sum_{j} \sum_{i} \delta_{ij} \left( c_{i}^{t-1} - x_j \right)^2
\]

3. Update cluster centers as the mean of the points

\[
c^t = \arg\min_c \frac{1}{N} \sum_{j} \sum_{i} \delta_{ij}^t \left( c_{i} - x_j \right)^2
\]

4. Repeat 2-3 until no points are re-assigned (\( t=t+1 \))
Nearest Neighbour Classification

NN and K-NN

• Classify using training data \((x_i, y_i)\)
• NN: Use the label of the nearest neighbour
• KNN: Use the label of the majority of the k nearest neighbours
• Regression: Use the average of the value of the k nearest neighbours
• Easy to implement and understand
• Can use arbitrary distance functions between images
• Converges to the optimum
• Slow when using lots of data, need to store all training data, not smooth regression
Nearest Neighbour Classification (discussion)
Parametric density estimation

Plug-in Classifier

- **Parametric density estimation**
  - (compare with Nonparametric density estimation)
    - Parametric – fixed nr of parameters
    - Nonparametric – nr of parameters grow with training data

- Plug-in classifier, i.e. plug-in the estimated densities in Bayes rule
- Classification
Logistic regression

- Linear logistic regression
- Estimate the posterior
- As linear function followed by standard logistic function

- Convex optimization problem

$$\min_w \frac{1}{2} w^T w + C \sum_{i=1}^{l} \log(1 + e^{-y_i w^T x_i}).$$

- Then classify according to

$$P(Y = y | X = x) = s(w^T x + b)$$
General Vector Space

• A 'General' Vector Space is a collection of objects called **vectors**, which can be added together and also be multiplied by 'numbers' called **scalars**, where the **addition** and **multiplication with scalars** fulfill a set of rules.

• There are many examples of such vectors spaces. The vectors can for example be
  • Geometrical vectors in three dimensions
  • N-tuples of real numbers
  • Functions
  • Polynomials
  • Matrices
  • Tensors
Example 3.2.2. Matrices of size $2 \times 2$ is a vector space. One possible basis is

$$\bar{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \bar{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \bar{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \bar{e}_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$  

The matrix

$$\bar{u} = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}$$

has coordinates $u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 7 \\ 2 \end{pmatrix}$, since

$$\bar{u} = u_1 \bar{e}_1 + u_2 \bar{e}_2 + u_3 \bar{e}_3 + u_4 \bar{e}_4 = \begin{pmatrix} 1 \\ 7 \\ 3 \\ 2 \end{pmatrix}.$$  

The dimension of the vector space is 4.
Linear Algebra

• Linear Algebra
  – Basis, coordinates
  – Scalar product
  – Projection onto a subspace
  – Projection onto affine subspace
  – Principal Component Analysis
  – Change of basis

• Vector space – ‘A matrix is a vector’ What does this mean?
  – All of the above applies to other vector space, e.g. space of images
What is the orthogonal projection of $f$ onto the space spanned by $(e_1, e_2, e_3)$?
Since \((e_1, e_2, e_3)\) is orthonormal, the coordinates are 
\[ x_1 = f \cdot e_1 = -2457, \quad x_2 = f \cdot e_2 = 303, \quad x_3 = f \cdot e_3 = -603. \]
The orthogonal projection is then \(\hat{f} = -2457e_1 + 303e_2 - 603e_3\).
3.2.3 Principal Component Analysis

In the previous two sections we talked about projection onto a subspace (defined by vectors $a_1, \ldots, a_k$) or onto an affine subspace (defined by vectors $m, a_1, \ldots, a_k$). But how can we determine a suitable subspace from examples.

The key idea here is that given many examples $x_1, \ldots, x_N$ in $\mathbb{C}^n$ or $\mathbb{R}^n$ find a subspace or affine subspace $\pi$ so that the errors when projecting all of the examples are small in some sense. The calculations become particularly easy if we choose a particular error.

Assume that an affine subspace $\pi = \{w | w = m + \sum x_i a_i = Ax + m\}$ where $x_i$ are examples in $\mathbb{C}^n$ (or $\mathbb{R}^n$).

The affine subspace $\pi$ that minimizes $e(\pi)$ can be found by the following method.

1. Calculate the mean $m = \frac{1}{N} \sum x_i$.
2. Subtract the mean from all examples $z_i = x_i - m$.
3. Place all of the resulting vectors as columns of a matrix, $M = (z_1 \ldots z_N)$.
4. Factorize $M$ using the singular value decomposition $M = USV^T$.
5. Use the first $k$ columns of $U$ as the basis of the subspace, i.e. $a_i = u_i$, with $U = (u_1 \ldots u_m)$.

3.2.4 Images as elements of a vector space

A digital grayscale image can be represented by a matrix $f = \begin{bmatrix} f(1,1) & \ldots & f(1,N) \\ f(2,1) & \ldots & f(2,N) \\ \vdots & \ddots & \vdots \\ f(M,1) & \ldots & f(M,N) \end{bmatrix}$.

Here we let the matrix elements $f(i,j)$ be complex (or real) elements. We will use the following notation to denote row $j$ and column $k$.

$f(j,\cdot) = [f(j,1) \ldots f(j,N)]$,

$f(\cdot,k) = [f(1,k) \ldots f(N,k)]$,

Introduce the following notation for column-stacking of a matrix.

$ef = \begin{bmatrix} f(\cdot,1) \\ \vdots \\ f(\cdot,N) \end{bmatrix}$. 

Review of Linear algebra
Fourier transform
Linear space, basis
Scalar product
Orthogonal projection
Illustration
Kalle Åström

Image Analysis - Lecture 2
Discrete Fourier Transform - 2D

\[
F(u, v) = \sum_{x=1}^{M} \sum_{y=1}^{N} f(x, y) e^{-i2\pi((u-1)(x-1)/M+(v-1)(y-1)/N)}
\]

\[
f(x, y) = \frac{1}{MN} \sum_{u=1}^{M} \sum_{v=1}^{N} F(u, v) e^{i2\pi((u-1)(x-1)/M+(v-1)(y-1)/N)}
\]
Any linear and translation invariant system can be represented as a convolution.

\[ g = f \ast h \]

\[ h = \delta \ast h \]
Convolution - repetition

• Convolution:
  - Flip the filter in both dimensions (bottom to top, right to left)
  - Then apply cross-correlation
  - Produces scalar product of flipped filter at every position!
Where’s Waldo?

Scene

Template
Template matching

\[(f \cdot 2) \ast e - 2f \ast \hat{h} + h \cdot h\]

```
e = ones(size(h))
hnorm2 = norm(h, 'fro')^2;
hhat = flipud(fliplr(h))
```

```
e =
1 1 1
1 1 1
1 1 1

hhat =
0 1 0
1 4 1
2 1 0
```

```
r = conv2(f.^2, e, 'same') - 2*conv2(f, hhat, 'same') + hnorm2
```

```
r =
31.0000 48.0000 40.0000 26.0000 25.0000
24.0000 54.0000 55.0000 48.0000 66.0000
52.0000 51.0000 52.0000 -0.0000 27.0000
42.0000 40.0000 88.0000 60.0000 54.0000
29.0000 38.0000 47.0000 22.0000 21.0000
```
The Convolution Theorem

\[ h = f \ast g \]

\[ H(u, v) = F(u, v)G(u, v) \]

\[ \text{FFT2}(f) \ast \text{FFT2}(g) = \text{FFT2}(\text{conv2}(f, g, 'periodic')) \]
Filter example: Differentiation and Gaussian in $x$-direction
Scale Space

Theorem
An operator $T_t$ with the following properties

- $T_t$ is a **linear and translation invariant** operator for every $t$,
- **Scale invariance.** If a function is scaled with a factor $\lambda$, i.e. $g(x) = f(x/\lambda)$ then there exists a scale $t' = t'(t, \lambda)$ such that $T_t g(x) = (T_t f)(x/\lambda)$,
- **Semi group property:** $T_{t_1}(T_{t_2} f) = T_{t_1 + t_2} f$,
- **Positivity preserving:** $f > 0 \Rightarrow T_t f > 0$,

is given by

$$T_t f = f \ast G_{\sqrt{t}}.$$
A bar in the big images is a hair on the zebra’s nose; in smaller images, a stripe; in the smallest, the animal’s nose.
Blob detection

- See matlab journal
- Local maxima – bad
- Convolute with Gaussian
  - Local maxima
  - Thresholding
  - Interpolation -> sub-pixel
Edge Detection
Convolve with “d/dx” and “d/dy”

Gradient Magnitude

Original image  Gradient magnitude  Non-maxima suppressed
Ridge detection

- Filter with elongated gaussians in different directions
Harris Corner Detector

Eigenvalue two of the orientation tensor

Two large Eigenvalues
Gives a corner
SIFT
(Scale Invariant Feature Transform)
Texture Detection

Convolution, filter \( h \)

\[
g(i, j) = \sum_u \sum_v f(i - u, j - v) h(u, v)
\]

Many convolutions, filterbank \( h \) with \( K \) filters

\[
g(i, j, k) = \sum_u \sum_v f(i - u, j - v) h(u, v, k)
\]