Neural Networks as Self-Learning Image Parameters

Patrick Reijnen

Catholic University of Nijmegen,
The Netherlands
University Hospital Nijmegen
Biophysics Laboratory
Department of Ophthalmology

Faculty of Mathematics and Informatics
Department of Real-Time Systems

Master's Thesis, number **329**
September 1994

Preface

This thesis is the result of my graduation project to receive my Master's degree in Computer Science at the Catholic University of Nijmegen. This project which ran from Februari 1994 to September 1994, was a cooperation of the Department of Real-Time Systems of the Faculty of Mathematics and Informatics and the Biophysics Laboratory of the Department of Ophthalmology of the University Hospital Nijmegen.

The goal of this project was to improve medical diagnosis. This goal is tackled by using neural networks to create new parameters which should be able to distinguish better between different kinds of echographic texture than the current statistical parameters.

At this point I want to thank dr. ir. Han Thijssen, ir. Henkjan Huisman and dr. Theo Schouten, my supervisors at the Biophysics Laboratory and the computer science department respectively, for their support, advice, help and - most of all - patience. Furthermore, I like to thank the other people of the Biophysics Laboratory for their support and help, everybody (and especially Peter Groeneveld) at Cafe "De Mark", my regular pub, for their willingness to hear my complaints when I met hard times during the project and last but not least my girl friend Petra Kipp for being there when I needed her.

Patrick Reijnen
Nijmegen, September 1994

Contents

Chapter 1

Introduction

At the Biophysics Laboratory of the Department of Ophthalmology research is done in ultrasound diagnostics. Current research is directed to images with local lesions.

From research and literature it is known that texture parameters provide extra information yielding increased distinction of lesions and normal tissue. The problem is that already very many texture parameters are known and that more parameters can be found which also increase distinction. From a practical and statistical point of view, only a small number of these parameters will be adequate for good distinction. A first strategy is to use those parameters which give the highest distinction.

In this research another way will be followed: existing parameters will not be used, but some new, maximally distinguishing parameters have to be derived from a training set, by training neural networks with this set. For each image pixel a local environment is presented as input to the neural networks. The neural networks are then trained to find the label belonging to this environment.

Questions to be answered are:

- What neural networks can be used to find new parameters?
- Can the neural networks find more information than is already contained in known parameters?

Research done to find satisfactory answers will be described in this thesis.

This research will fit into the following framework:

In chapter 2 background theory is given. As this research is concerned with ultrasound, it will be described what ultrasound is, what is needed to transmit and receive ultrasound and how echoes received can be visualized.

Furthermore, texture and speckle are explained, as these contain the information wanted.

After the presentation of the background theory it will be possible to present a problem definition in chapter 3. First, tissue characterization will be described. Then, texture parameters in use are presented and some problems of these parameters are mentioned. Last, an explanation will be given why neural networks will be used to find new parameters.

Chapter 4 comprises a literature study on neural networks. This study has to cast some light on what neural networks can be used to find the new parameters. Results of neural networks in all kinds of applications will be mentioned to find problems, advantages and disadvantages of these networks. Then problems, advantages and disadvantages are compared and a conclusion drawn on what neural network(s) will be used in this project.

Chapter 5 describes how the first tests are done with the neural networks chosen. These tests are carried out to check if the neural networks chosen are really capable of finding new parameters.

In chapter 6 more tests are described. The aim of these tests is to find out whether the neural networks chosen will or will not give better results than the known parameters.

Next, in chapter 7 the neural networks are used on more realistic, one-dimensional, simulated images. The parameters used in generation of these images will be presented with their values. Then several tests and their results will be described. The aim of these tests still is to find out what neural networks will give better results than the already known parameters. Another aim of the tests will be to check the tests done so far on influences not looked at and on test methods not used so far.

Finally, conclusions and recommendations for further research are laid down in chapter 8.

Chapter 2

Background theory

2.1 Introduction

This chapter deals with the theoretical background for this thesis, which is necessary to formulate a clear and understandable problem definition in chapter 3.

In the first section the science of acoustics is described, which science carries out research into all the aspects relevant to sound.

In section 2.3 the reader is made familiar with transducers, which are used in ultrasound diagnosis of tissues. The properties of transducers are explained as they are relevant for the problem definition.

Section 2.4 gives some background on the properties of waves in biological tissues. These properties are relevant to the problem definition as they have much influence on the information contained in the echo signal received.

In section 2.5 the A- and B-mode imaging methods are explained, which can be used to visualize the echoes received. The way of visualization gives us more or less information on the anatomy and structure of the tissue scanned.

Finally, section 2.6 describes what is meant by texture and speckle, how they arise and how they can be used for medical diagnosis. Also, first-order and second-order speckle statistics are dealt with.

2.2 Acoustics of waves

The science of acoustics is the science of sounds. It comprises all mechanical waves of all possible frequencies and in all possible materials. A mechanical wave is defined as follows:

> A mechanical wave is a systematic disturbance of the condition of rest of a material. The position of this disturbance in space changes in time. **Mechanical wave**

There are two kinds of waves:

1. Transverse wave **Transverse wave**
   This wave is propagated perpendicularly to the direction of motion of the particles of the material in which the wave is propagated.

2. Longitudinal wave **Longitudinal wave**
   The propagation direction of this wave is the same as the direction of motion of the particles of the material in which the wave is propagated.

Waves can be characterised by the following physical parameters:

- Propagation speed (c):
  The speed at which the wave is propagated through the material. This speed depends on mechanical properties such as density and elasticity.
- Vibration frequency (f):
  The number of times per second particles complete a movement cycle.
- Vibration amplitude:
  The maximum lateral movement which can be realised starting in a state of rest.
- Wave length (λ):
  The wave length is the constant distance in the propagation direction which separates particles with the same deviation. Wave length can be computed with the formula:

$$\lambda = \frac{c}{f}$$

The working area of the science of acoustics is divided into frequency areas:

- Infrasonics: sound with frequencies which are too low to be heard by humans.
- Audiosonics: sound that can be heard by humans.
- Ultrasonics: sound with frequencies which are too high to be heard by humans.

2.3 Transducers and ultrasound waves

In medical diagnostics transducers are used for transmitting and receiving ultrasound. The material of which transducers are made needs to have piezo-electrical properties. With the aid of the piezo-electrical effect acoustic pressure waves received can be transformed into voltages and voltages can be transformed into acoustic pressure waves. Voltages on the piezo-electrical material cause changes to the thickness of the material. This property is used to create sound vibrations. The ideal transducer is efficient as a transmitter and sensitive as a receiver.

Some properties of transducers are relevant to the resolution:

Lateral resolution

1. Lateral resolution
   Lateral resolution is measured by moving the transducer in the lateral direction. During the passage of two reflectors at the same depth/distance the strength of the echo received first increases and then decreases until a second reflector is reached which again causes the strength to increase. If two small dots can just be seen in an image, the distance between the reflectors corresponds with the lateral resolution of that transducer (figure 2.1, "4").

   For echo signals reflected by reflectors at the same distance L from the transducer no distinction can be made in the width of the beam as to the places the echoes are coming from (figure 2.1, "3").

   In order to get a better lateral resolution the transducer needs to be focussed.

Axial resolution

2. Axial resolution
   With axial resolution it is tried to distinguish reflectors separated in the axial direction, which is the direction of the axis of the sound beam (figure 2.1, "1").

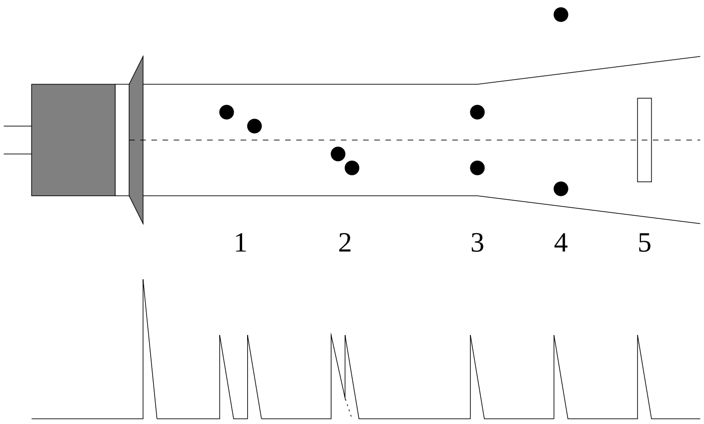

Figure 2.1: A-mode image

The time duration of the echo pulse is important when a clear distinction is wanted between two structures in depth. The echo pulses received may not be allowed to overlap completely (figure 2.1, "2").

For a good axial resolution a short sound pulse is needed. To create this short sound pulse a transducer is needed which has a frequency bandwidth which is big enough.

The intensity of sound transmitted by a transducer is not the same in every direction. It is transmitted into a specific direction. The width of this beam normally depends on the distance to the transducer. A distinction is made between the near-zone and the far-zone (figure 2.2).

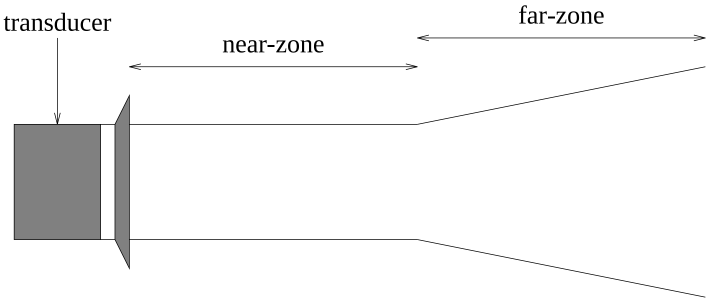

Figure 2.2: Sound field of a flat transducer

Near-zone

The near-zone (Fresnel-zone) is the sound field at a short distance from the transducer in which interference takes place. The length of the near-zone is determined by the dimensions of the active surface and the wave length used. If each surface particle of the transducer is seen as a small transmitter, then in a point P in the near-zone all contributions are added together causing interference.

The diameter of the sound beam in the near-zone is almost equal to that of the transducer. Sound vibrations of different pieces of the transducer arrive at one point with in-between time differences. The vibrations amplify or attenuate each other (interference). Time differences in the near-zone can cause strong extinction in various places, which takes place when the corresponding differences in path are one or more times $\frac{1}{2}\lambda$ (λ is the wave length).

Far-zone

The far-zone (Fraunhofer-zone) is the field in which for a point R the path differences are smaller than $\frac{1}{2}\lambda$ between border and centre of the transducer. Therefore, in the far-zone no extinction will occur in the main beam. In the far-zone the width of the beam grows proportionally to the distance.

The change from near-zone to far-zone takes place at the point where the difference in wave length between border and centre of the transducer is exactly $\frac{1}{2}\lambda$.

Diffraction

Diffraction occurs when the waves are bent due to the fact that a finite size aperture is used (transmitting and receiving surface). Because the aperture is of a finite size beside plane waves also spherical waves occur at the edges of the aperture.

Side lobe

Outside the main beam, secondary beams may occur due to diffraction. In these beams secondary maxima, also called side lobes, may occur due to interference of wavelets. The place of the side lobes depends on the transducer frequency chosen, the diameter and form of the surface of the transducer. If a hard reflector is situated in the place of a side lobe a "ghost reflector" is created in the image.

2.4 Properties of waves in biological tissues

Biological tissues have an inhomogeneous composition, where inhomogeneity is defined as a spatial change of acoustic properties:

Reflector

- The size of the inhomogeneity is large with respect to wave length. Such a type of tissue is called a reflector.

Scatterer

- The size of the inhomogeneity is smaller than or equal to the wave length. Such a type of tissue is called a scatterer.

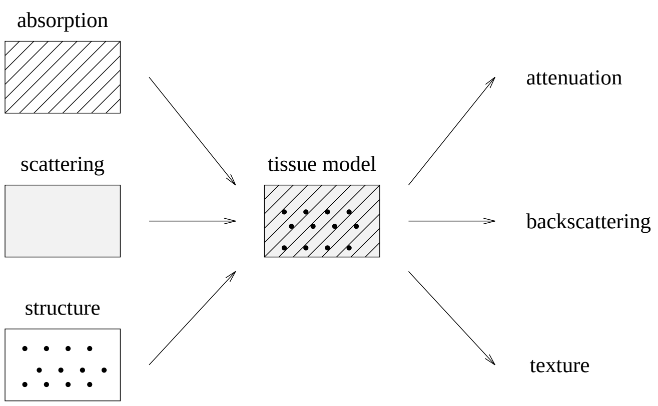

Figure 2.3: Acoustic tissue model

The biological tissue model for echographic signals (figure 2.3) consists of a combination of reflectors and scatterers in a homogeneous medium. Biological tissues have different physical properties which can be observed by means of ultrasound. Three of these properties are:

Absorption

- **Absorption of ultrasound**
  Absorption of ultrasound is mainly caused by relaxation phenomena of biological macromolecules, which transfer mechanical energy into heat.

Scattering

- **Scattering of ultrasound**
  Scattering may be omnidirectional or directional, due to an inhomogeneity which may be best described as a secundary source in a homogeneous medium.

When receiving the backscattering from a dense cloud of small scatterers, interference will result on reception by the transducer. When the sound field is moved with respect to the cloud the interference pattern remains coherent over +/- half the beam diameter, i.e. a speckle pattern becomes visible. The size of the speckles is dependent on the transducer characteristics and on the number density of the (random) scatterers. **Speckle**

- **Structure** **Structure**
  In addition to randomly positioned scatterers there may be a second class of scatterers in the tissue, which is regularly spaced. This structure may be resolved or unresolved, meaning that the distance between scatterers is too small or large enough respectively to distinguish between separate scatterers.

From the tissue model the following measurements can be derived:

- **Attenuation of ultrasound** **Attenuation**
  Attenuation is defined as the total loss of energy per unit length from the ultrasound wave as it is propagated through tissue. In pulse-echo imaging, there is a reduction of the ultrasound wave amplitude due to reflection, scattering, diffraction, refraction, beam divergence and absorption.

  Attenuation can be measured either by through transmission, by the decrement of echo amplitude with increasing range.

  Due to attenuation the echo amplitude of structures deeper in the material will be lower than the echo amplitude of those closer to the transducer. Echoes of the structures deeper in the material will be received later in time than echoes of structures closer to the transducer. One method to compensate for attenuation is called Time Gain Compensation. With this method echo signals received early in time are amplified slightly and echo signals received later in time are amplified more. In this way all the echo signals have more or less the same strength which is needed to anticipate saturation and loss of echo signals in the echogram. **Time Gain Compensation**

- **Backscattering** **Backscattering**
  Backscattering is a special form of scattering. The term backscattering is used for that part of the scattering echoes which has a degree of 180^o with respect to the direction in which the beam is transmitted. The backscattered energy is the only information which can be displayed in an image of a scattering medium.

- **Texture** **Texture**
  See section 2.6

2.5 B-mode imaging of backscattered waves

There are a number of ways to display an echo signal. The B-mode method is the one used for generation of the images. As a B-mode image is generated from several A-mode images first the A-mode method will be explained. An A-mode image results if the envelope of the radiofrequency (RF-) signal is determined. After that the demodulated signal is compressed and filtered. The time signal which is created in this way is called the A-mode (Amplitude-mode) echogram. **A-mode**

If the sound speed of the tissues the ultrasound is transmitted through is known, the distance can be calculated in millimeters of tissue between the transducer (on the human body) and the reflector in the body by using the time-of-flight.

B-mode

The second method is to display the B-mode image. The B-mode image is created from a set of succesive A-mode echograms obtained by scanning the beam in a single plane. By displaying the amplitude as a brightness and displaying the A-mode echograms side-by-side on a monitor a two-dimensional B-mode (Brightness-mode) echogram is created.

There are three aspects of resolution which are relevant to (ultrasound) imaging:

Spatial resolution

1. Spatial resolution describes the ability of an imaging system to produce in the image distinguishable registrations of closely positioned structures within the patient.

Contrast resolution

2. Contrast resolution is a measure of the ability to produce in the image displayed differences in the brightness of the registrations of structures with slightly different echogeneities within the patient.

2.6 Texture and speckle in B-mode images

Speckle

B-mode scans of pathological tissues may show a texture different from that in B-mode scans of normal tissues. Texture differences form a regular basis for clinical diagnostics. A simplified model for the generation of tissue textures ([Thijssen, 1992]) from scattering media is as follows (figure 2.4): a transducer insonating a homogeneous medium that contains four point-like scatterers. These scatterers yield spherical waves that will arrive at the transducer at slightly different times after the transmission of the ultrasound pulse. The transducer produces an electrical signal that is the algebraic sum of the instantaneous sound pressures originating from the four backscattered waves. The four echoes will build an interference pattern. The texture obtained in this way has a granular structure, for which the term "speckle" is used. Speckle presents itself in an image as a "grain" with high contrast.

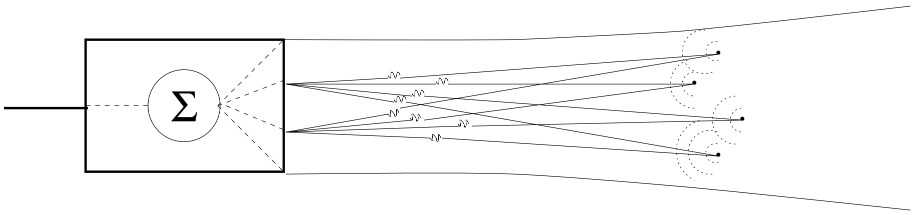

Figure 2.4: Scattering by small structures yields spherical wavefronts

The speckle pattern in an image does not have a one-to-one correspondence with echo-producing targets. Rather is it an artifact that depends on characteristics of the tissue scanned and the imaging system.

Speckle can be wanted or unwanted. When you are only interested in anatomical details speckle is an unwanted disturbance. For relatively homogeneous tissues like orbital fat the B-image contains only a speckle pattern. This pattern can be used to find the properties of the tissue scanned.

2.6.1 Speckle statistics in theory

When the number of scatterers in the resolution volume of a coherent imaging system is very large, i.e. above a certain practically determined number (ten), the statistical properties of the speckle pattern are independent of the number of scatterers. In that case these properties only depend on the physical dimensions of the transducer.

Another extreme can be found in the case the number of scatterers in the resolution volume decreases to zero. Then the B-mode echogram does not show much speckle.

Between these two extremes the statistical properties of the speckle pattern depend on the number of scatterers and properties of the imaging system. When the resolution volume contains enough (more than ten) randomly distributed scatterers the speckle is called fully developed. For fully developed speckle statistical properties can be computed.

Fully developed speckle

First-order statistics

If the number of scatterers within one resolution element is large and the phases are distributed uniformly between 0 and 2π, the amplitude obeys a Rayleigh probability density function $p(A)$ ([Burckhardt, 1978])

$$p(A) \;=\; \tfrac{A}{\psi}\, exp\,(-A^2 \,/\, 2\psi)$$

where ψ is a parameter. It can be shown ([Burckhardt, 1978]) that A has a rather large probable variation around its average.

The average amplitude μ_A and its standard deviation, which will be denoted as noise amplitude σ, are determined as follows ([Burckhardt, 1978]):

$$\mu_A \;=\; (\pi\psi \,/\, 2)^{\frac{1}{2}}$$

$$\sigma \;=\; \left[4\psi^2 \;-\; \tfrac{\pi\psi}{2}\right]^{\frac{1}{2}}$$

The signal-to-noise ratio (SNR) is now defined as ([Burckhardt, 1978], [Thijssen, 1992])

$$SNR \;=\; \tfrac{\mu_A}{\sigma} \;=\; \frac{\mu_A}{[\;\overline{A^2} - \mu_A^2]^{\frac{1}{2}}}$$

For fully developed speckle the SNR has the value $(\frac{\pi}{4-\pi})^{\frac{1}{2}} \;=\; 1.91$.

Second-order statistics

Second-order statistics are described by the autocovariance function (ACVF). The ACVF can be derived theoretically for fully developed speckle caused by randomly distributed point scatterers in the focal zone of the transducer.

Using the ACVF an average Full-Width-at-Half-Maximum (FWHM) in axial and lateral directions can be found ([Thijssen, 1992]).

The FWHM for the lateral direction is

$$FWHM_x \;=\; 0.86\lambda_c F_g / D \tag{2.1}$$

For the axial direction the FWHM is

$$FWHM_z \;=\; 2.17\sigma_z \tag{2.2}$$

where λ_c is the wave length belonging to the central frequency,
F_g is the geometrical focussing distance,
D is the geometrical diameter of the transducer,
σ_z stands for the standard deviation of the ultrasound pulse in the axial (z-) direction.

2.6.2 Factors influencing speckle statistics

Factors which influence speckle statistics can be subdivided into imaging system dependent factors and tissue dependent factors.

Imaging system dependent factors

The speckle pattern partly depends on the frequency, bandwidth, diameter and focussing degree of the transducer of the imaging system as can be seen from Eq. (2.1) and (2.2). The size of speckles is given by the autocorrelation function which depends on performance characteristics and dimensions of the transducer employed. The axial speckle size is exclusively and inversely dependent on the bandwidth of the transducer employed, i.e. proportional to the spatial pulse length. The lateral speckle size is proportional to the focal length and inversely proportional to the transducer diameter and to its frequency.

Diffraction effect

The speckle characteristics are strongly dependent on the distance between the transducer and the scatterers. This depth dependence is called the diffraction effect. The diffraction effect has an influence on the mean and the SNR of the speckle and the FWHM of the ACVF ([Oosterveld, 1984], [Fabel, 1988]). The influence of the distance on the ACVF is mostly on the lateral FWHM. The lateral FWHM increases with the distance to the transducer. Behind the focussing point it levels off. The axial FWHM is almost constant.

Tissue dependent factors

Attenuation

Attenuation and scattering density and structure influence the tissue texture. Attenuation by a tissue has no influence on the axial size of speckle, whereas the lateral size increases proportionally to the magnitude of the attenuation coefficient, but also to the penetration depth.

Scattering

In case of diffuse scattering and a low number density of the scatterers the speckle size is larger than in the high density case in which fully developed speckle will occur.

Structure

[Jacobs et al., 1991] and [Wagner et al., 1987] state that a structural component in a tissue gives rise to "distributed specularity" when the spacing distance of the scatterers is smaller than the wave length. Compared to the purely random walk this results in a significant increase of the SNR.

In case of fully developed speckle due to diffuse and structural scattering the SNR has a value larger than 1.91 (distributed specularity) for unresolved scattering. When the structural scatterers are resolved the SNR has a value smaller than 1.91.

Chapter 3

Problem Definition

3.1 Tissue characterization

Ultrasonic tissue characterization deals with ultrasound-tissue interaction and can be defined as an attempt to establish a relationship between a quantitative parameter from an acoustic measurement on a tissue and the histology of that tissue. The general purpose of tissue characterization is to increase the sensitivity and specificity of medical diagnosis by classifying a tissue on the basis of quantitative measurement.

In this thesis two classes of parameters are distinguished:

1. parameters related to a physical phenomenon
2. parameters not related to a physical phenomenon

The first class of parameters is tested in tissue characterization, because there is a direct link between tissue histology change and a change in some physical phenomenon. By measuring this phenomenon, information should be obtained about the tissue.

The second class of parameters consists of existing image analysis methods or any other method that results in a parameter that is sensitive to tissue histology changes.

In this thesis it is the second class of parameters that will be focussed on. It is known that the spatial characteristics within an ultrasound image region play an important role in tissue discrimination ([Thijssen et al., 1990]). Therefore in this project, texture parameters will be analysed. In section 3.2 a survey is given of some of the most often tested parameters.

In section 3.3 problems with and restrictions of the parameters used are presented and finally, in section 3.4 a description is given of the way tissue characterization is applied in this thesis.

3.2 Texture parameters in use

The texture parameters used for tissue characterization can be derived from several second-order grey-level statistical methods three of which will be mentioned here:

- The Autocovariance Function
  The first class of texture parameters can be derived from the autocovariance function. This function and some of the parameters derived from it have already been mentioned in section 2.6.1.

Autocovariance Function

Co-occurrence matrix

- Co-occurrence matrix
  The second class of texture parameters is that derived from the co-occurrence matrix.

  The co-occurrence matrix $C_{\vec{d}}$ (grey-level spatial-dependence histogram) is a two-dimensional histogram characterizing the occurrence of grey-level combinations in pairs of spatially related pixels. The matrix entry $C_{\vec{d}}\,(i,j)$ thereby specifies how often two pixels which are separated by a given displacement vector in the axial and lateral directions $\vec{d} = (dx, dy)$ display grey levels i and j respectively ([Haralick et al., 1973], [Gerbrands, 1989], [Räth et al., 1985], [Nicholas et al., 1986]).

  Before statistical properties are obtained from the co-occurrence matrix, the matrix entries are normalized with respect to the total number of entries. This creates a matrix for a (dx, dy) - transition:

  $$P_{\vec{d}} : [\,p(i,j)]$$

  Where

$p(i,j)$	$=$	$C_{\vec{d}}(i,j)\;/\;R$
$\sum p(i,j)$	$=$	1
R		is the number of neighbouring resolution cell pairs used in computing the particular grey-tone spatial-dependence matrix.

  Several characteristic features can be extracted from $P_{\vec{d}}$:

Contrast

  - **Contrast**

    $$CON \;=\; \sum_{i,j} (i-j)^2 * p(i,j)$$

    CON characterizes the contrast in the region of interest. Frequently occurring large grey-level differences increase the CON-value, whereas soft texture results in small CON-values.

Angular second moment

  - **Angular second moment**

    $$ASM \;=\; \sum_{i,j} p(i,j)^2$$

    ASM is a measure of homogeneity of an image. ASM increases when the co-occurrence matrix values cluster around a major grey-level transition. This corresponds to a situation where only a small number of different grey-level transitions exist. ASM is not dependent on the mean grey level.

Entropy

  - **Entropy**

    $$ENT \;=\; -\sum_{i,j} p(i,j) * \log\; p(i,j)$$

ENT measures the homogeneity or uniformity of the $p(i,j)$-value, and therefore it increases with increasing coarseness of the image texture. ENT is not dependent on the mean grey level.

– **Correlation**

Correlation

$$COR = \frac{[\sum i * j * p(i,j) - m_x * m_y]}{\sqrt{s_x^2 * s_y^2}}$$

Using

$$m_x = \sum_i i \sum_j p(i,j)$$

$$m_y = \sum_j j \sum_i p(i,j)$$

$$s_x^2 = \sum_i i \sum_j p(i,j) - m_x^2$$

$$s_y^2 = \sum_j j \sum_i p(i,j) - m_y^2$$

COR measures the linearity of the relationship of the grey levels in d-related pixels.

- Run-length statistics

Run-length statistics

The run-length statistics belong to the third class that will be described here ([Haralick, 1979], [Conners et al., 1980]).

The two-dimensional grey-level run-length histogram counts the number of grey-level runs by their length and grey-level range. A run is a set of neighbouring pixels in a certain direction displaying similar grey levels.

The count distribution of the histograms can be characterized by the following parameters. In these parameters

N	is the number of pixels in the region of interest
$GRHIST(l,g)$	is the number of runs of length l and grey level g in the region of interest

– **Run percentages**

Run percentages

$$RP = \frac{1}{N} * \sum_{l,g} GRHIST(l,g)$$

RP characterizes the distribution of runs; this parameter increases with increasing inhomogeneity of the texture. In a homogeneous image, when only a small number of runs is present, the run percentages are small.

Long-run emphasis

– **Long-run emphasis**

$$LRE = \frac{1}{N} * \sum_{l,g} l^2 * GRHIST(l,g)$$

The long-run emphasis LRE is largely dependent on the occurrence of long runs.

Grey-level distribution

– **Grey-level distribution**

$$GLD = \frac{1}{N} * \sum_{g} \left[\sum_{l} GRHIST(l,g) \right]^2$$

GLD characterizes the grey-level uniformity of the runs. A texture pattern displaying uniformly coloured grey-level runs will be characterized by a high GLD value.

Run-length distribution

– **Run-length distribution**

$$RLD = \frac{1}{N} * \sum_{l} \left[\sum_{g} GRHIST(l,g) \right]^2$$

RLD expresses the length uniformity. A texture pattern displaying uniformly spaced grey-level runs will be characterized by a high RLD value.

3.3 Problems with texture parameters in use

Research done on texture parameters in ultrasound tissue characterization shows varying statistical results.

[Layer et al., 1990] and [Pinamonti et al., 1989] state that good results can be obtained with the first-order parameter mean grey level (not described in the previous section). [Layer et al., 1990] also tested texture parameters but these did not result in extra information. As is said by [Pinamonti et al., 1989] and [Ferdeghini et al., 1991] entropy gives good results, while [Layer et al., 1990] shows that the results obtained for entropy are not so good. Good results are obtained by [Pinamonti et al., 1989] and [Ferdeghini et al., 1991] using the angular second moment or contrast parameters. [Schlaps et al., 1987] and [Nicholas et al., 1986] show good results by using the correlation parameter. Finally, combinations of parameters can be used. [Layer et al., 1991] show that marginally better results can be obtained by using fifteen texture parameters.

All the previously mentioned researchers have chosen one or more parameters out of the many parameters available. This was necessary because the data sets available were small while the precision of the tests had to be high enough. They had to find the best parameters for their examinations without discarding too much information. A problem in this case is that they will never know if the parameter or the combination of parameters they have chosen will be the best, as the size of the set of possible parameters and their combinations is too large to test them all.

Finding the best parameter or set of parameters which discards as little information as possible is difficult due to the following problems:

- Standardized examination conditions have to be used. As these can perhaps not be reached by other researchers, the results obtained are less useful. Furthermore, due to the standardized conditions some information is lost.
- Depth, diffraction and other dependencies have to be corrected as well as possible. As these dependencies cannot be fully corrected information specific for the measurement conditions is partly lost.
- Many researchers only present a percentage of accuracy of the texture classification. Results of sensitivity and specificity, although very important for clinical diagnostics, are hardly ever provided. Therefore, it is not possible to see whether the results of the parameters chosen are really useful or completely useless.

3.4 Finding new texture parameters

As can be seen, finding the best parameters is a hard job due to the problems met with.

In this thesis a method will be described giving one or more new texture parameters which have to overcome some of the problems mentioned in the previous section. This method will extend the set of currently available parameters (figure 3.1).

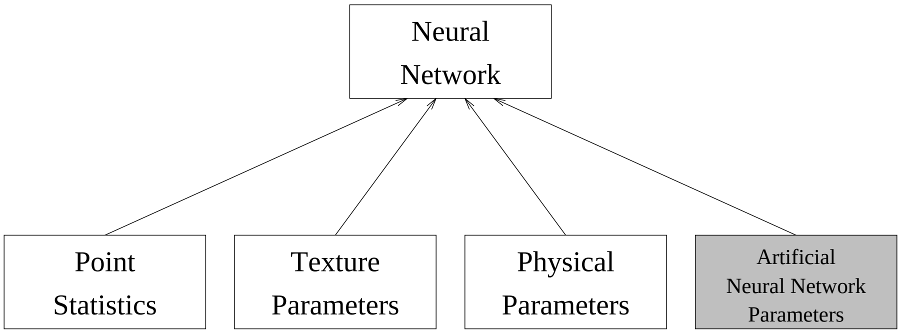

Figure 3.1: Current state in NN ultrasound research and new extension

The parameters created with this method will fall in a new category of parameters called 'Artificial Neural Network Parameters' (the shaded box in figure 3.1).

The new method is based on a neural network. This neural network will have to be learned how it has to make a new, maximally distinguishing parameter or set of parameters from the image information presented to the network. The image information consists of a local environment of each image point. For this environment the neural network will learn the label belonging to it. Problems mentioned in the previous section will also influence the image information used to train the neural network.

One advantage of this new method will be that the neural network is capable of finding parameters which can be represented by a non-linear function of the input. This function is composed of the weight functions of the network layers. Following this method the parameter or set of parameters found may yield more precise results.

The new method will be more flexible due to the fact that each parameter will be represented by a function being a combination of weight functions and not by a predefined linear or non-linear function. These weights fully depend on the image information offered to the network. Advantage of this might be that the parameter or set of parameters found is more sensitive than the parameters based on predefined functions.

Last, for each kind of texture one and only one parameter will be learned which can then be used to train another neural network for classification of unseen images of this kind of texture. No more choices have to be made like what and how many parameters have to be used for classification purposes.

These advantages will ensure that less information will be discarded than with the texture and physical parameters and that sensitivity, specificity and accuracy of texture classification will increase.

Disadvantages are that we do not know at the moment whether new parameters will be found or not; a parameter learned for one kind of texture will almost certainly not give good results for another kind of texture; so for every kind of texture a parameter has to be learned; providing badly preprocessed input data for a kind of texture to the neural network presents an unusable parameter for that kind of texture, which is not a problem with a predetermined, fixed set of parameters as the parameters will be correct although the possibility exists that they will not work for a specific kind of texture.

In the next chapters of this thesis the following questions will be answered:

- What neural networks can be used for these applications?
- Will the neural network be able to find more information than the existing parameters?

Chapter 4

Application of neural networks in image analysis

4.1 Literature survey of pattern and texture classification

In this chapter pattern and texture classification with neural networks will be described. This will be done following the three levels described in figure 4.1.

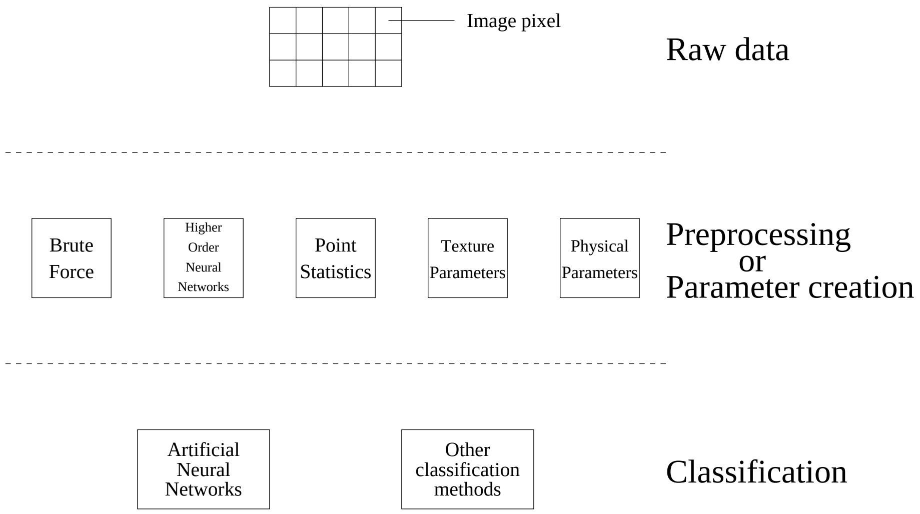

Figure 4.1: From raw image data to classified data

Raw data

The first level is the raw image data level.

Preprocessing

The second level is the preprocessing level. As this chapter is concerned with preprocessing methods using neural networks, only these methods will be highlighted in section 4.3. Furthermore, examples of the use of each neural network method for image analysis, as far as available, are given.

Classification

Finally, after the preprocessing part, at the third level classification can take place using Artificial Neural Networks, Discriminant Analysis or any other method available for classification. Section 4.4 will be focussed on the neural networks used for classification.

4.2 The raw data part

The raw data can be digitized versions of SAR images, echograms, normal photos or any other form of image. Depending on the quantization of the original scene, and many other factors, the raw data are a representation of the original scene, most certainly of a worse quality than this original scene. To improve the quality of the images and make them useful for diagnostic means, the image data are processed before being analysed.

4.3 The preprocessing part

As [Lippmann, 1989] mentioned, application of a pattern or texture classifier first requires a calculation of features that must be tailored separately for each problem domain. These features should contain information required to distinguish between classes, be insensitive to irrelevant variability in the input, and also be limited in number to permit efficient computation of discriminant functions and to limit the amount of training data required. Good classification performance requires calculation of effective features.

Preprocessing

Feature calculation is done in the preprocessing part. In literature it can be found that there are three ways to calculate features. The first way is to directly feed the raw image information to a neural network, which has to learn an unknown relation present in the image. This method is called "Brute Force".

Brute Force

This application of neural networks does not specifically take into account the spatial relations between the input data (i.e. raw pixels are related because of their topology). Another type of networks capable of training higher-order relations between input data is the HONN (Higher-Order Neural Network). It will be shown that this network for generating texture parameters is in fact a minimal form of preprocessing followed by a neural network.

HONN

Statistical and/or Physical

Finally, features can be calculated using statistical and/or physical parameters. As a description of this method has already been given in the previous chapter, it will not be mentioned here any further.

In the next paragraphs only the methods in which neural networks are used, are discussed.

Brute Force

Brute Force

With the Brute Force method, raw image data are directly fed into the input neurons of the neural network. In supervised training also the output patterns wanted are known. During the training phase, the network tries to decide which input patterns belong to which output patterns. It does so by giving weights to the connections between neurons in all layers. As supervised training is used, it is known before training which input pattern belongs to which output pattern. The neural network will have to learn this relation between input and output patterns.

[Windsor et al., 1993] have used the receptive field multilayer perceptron to which they directly fed 7x7x7x2 pixel ultrasonic images of defects in a weld, measured with an ultrasonic probe. Furthermore, they used the receptive field backpropagation method with four hidden units.

The adaptive receptive field method gave the best results with quite small receptive fields, typically just 3x3x3x2 pixels representing the depth, distance from weld, distance along weld and angle axes respectively. On a leave-one-out basis a success rate of 93.9% was achieved. This result was not dependent on the fraction of the data set used for training.

The receptive field backpropagation method also gave good results for small receptive field sizes. With a field of 3x3x3x2 pixels and four hidden units the leave-one-out performance was 94%. A shared weight network containing four receptive fields each of size 5x5x5x2 pixels gave 90% success in a leave-one-out analysis of the dataset.

[Silverman et al., 1990] used the backpropagation network to detect tumors (in eyes) in ten 128x128 8-bit pixel, linear-coded grey-scale ultrasonic images. The simulated network consisted of a 9x9 pixel input layer, two hidden layers of ten units each, and a single output unit. During processing, the 9x9 input layer of the neural network was stimulated by successive local 9x9 regions of the input image. Positional and scale properties were brought into the network by translating the network across the image and exposing it to the image at different scales.

For classification [Silverman et al., 1990] applied a training set of 138 tumors (in eyes) to backpropagation networks of various architectures:

- No hidden layers
- Single hidden layer having two modifiable layers of connections
- Two hidden layers having three modifiable layers of connections

Each network had four output units that corresponded to the four possible classifications. The sizes of the hidden layers were varied from 4 to 20 units. Five thousand iterations of the training set were run for each configuration.

The network performance was judged by the magnitude of the error, the percentage of correct classification and the convergence time, which was defined as the number of iterations required until the error was within 5% of the ultimate error.

For one hidden layer and melanoma subclassification the percentage of correct classifications was between 83.3% and 87.0%, without melanoma subclassification between 95.7% and 96.4%. Convergence times were between 1800 and 3500 passes. For two hidden layers these numbers are: between 71.0% and 100.0% with melanoma subclassification, between 95.7% and 100.0% without melanoma subclassification and from 1800 up to more than 5000 passes. These results were for a network which first learned all 138 cases and then classified them.

Another experiment done was to estimate how well backpropagation networks can recognize cases not included in the training set. From the set of 138 cases they left out 10 randomly selected ones. For one hidden layer and melanoma subclassification the percentage of correct classifications for cases in the training set was between 73.4% and 86.0%, without melanoma subclassification between 88.5% and 95.9%. With the unknown cases these results are between 71.1% and 76.9% and between 85.3% and 90.0% respectively. For two hidden layers the numbers for cases in the training set are: between 78.1% and 95.0% with melanoma subclassification, between 89.9% and 99.0% without melanoma subclassification. For the unknown cases the numbers are between 71.1% and 73.1% and between 86.6% and 89.1% respectively.

They found the predictive accuracy of the single-hidden-layer network superior to either the two-hidden-layer network or linear dicriminant analysis, although two hidden layers provided nearly perfect retrospective classification.

[Cho et al., 1992] did two experiments. For both experiments they used 990 printed Hangul syllables (Korean script). After scanning of the printed syllables, from each syllable a 40x40 binary image was created. For the first experiment they implemented a single backpropagation neural network with a single hidden layer which had 1600 (40x40) input nodes, 40 hidden nodes and 10 output nodes. The output nodes encoded distributively Hangul syllables. This network, however, did not converge when the number of classes of training examples exceeded about one hundred syllables.

For the second experiment they utilized the fact that a Hangul syllable is constructed by a combination of two to six alphabets. A syllable is grouped into six classes according to the types of alphabet combination. As a result, the overall classifier is composed of a global type classification network and six recognition networks, one for each global type (figure 4.2).

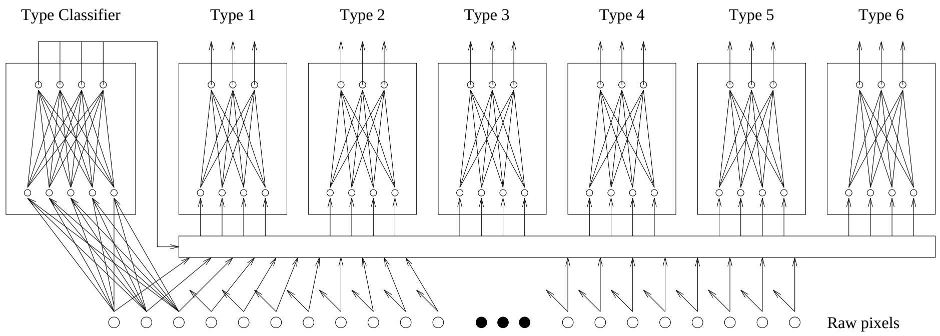

Figure 4.2: Neural network classifier to Hangul syllable recognition

The recognition rate for the first stage was 99.29%, for the second stage 96.06%. Advantages of the two-stage neural network classifier over a single backpropagation network are:

1. Higher recognition performance with much fewer training data.
2. Shortening of training time.

Higher-Order Neural Networks

Higher-Order Neural Networks

A high-order neuron can be defined as a high-order threshold logic unit which includes terms contributed by various high-order weights. A higher-order neural network is defined as a collection of high-order logic units ([Giles et al., 1987]).

Following [Reid et al., 1989a] and [Reid et al., 1989b] the output of nodes in a general higher-order neural network is given by:

$$\begin{aligned} y_i(p(x_p)) &= \Theta(\sum_j w_{ij} p(x_p)_j + \sum_j \sum_k w_{ijk} p(x_p)_j p(x_p)_k \\ &+ \sum_j \sum_k \sum_l w_{ijkl} p(x_p)_j p(x_p)_k p(x_p)_l + \cdots) \end{aligned} \tag{4.1}$$

where:

Θ	is a non-linear threshold function.
$p(x_p)_j$	is the value of a signal function p (sometimes called a pixel) at position j of position vector x_p. The values of the function p are used as exitation values of the input nodes.
w_{ij}	the interconnection matrix elements determine the weight that each input is given in the summation.
$y_i(p(x_p))$	is the value of output neuron i.

A diagram of a neural network utilizing only second-order terms is shown in figure 4.3. When a perceptron network is used the second-order update rule can be expressed as:

$$w'_{ijk} \;=\; w_{ijk} \;+\; [t_i(p(x_p)) \;-\; y_i(p(x_p))]\, p(x_p)_j p(x_p)_k$$

where:
$t_i(p(x_p))$ is the target output
$y_i(p(x_p))$ is the actual output

of the ith unit for input vector $p(x)$.

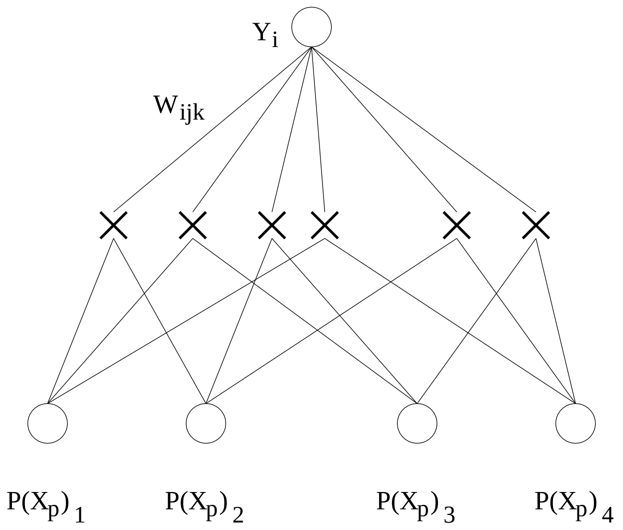

Figure 4.3: A second-order neural network with 4 inputs and 1 output

In higher-order neural networks it is possible to handcraft the units in such a way that their output is invariant under the action of an arbitrary finite group of transformations on the input space. Geometric invariances are a subset of this group. Geometric invariance means that the input space can be translated, scaled, rotated etc., without this action having effect on the output of the units of the HONN. Output of the units for the normal input space as well as for rotated, translated and scaled versions is thus the same ([Giles et al., 1987]). The invariances achieved require no training and apply to any input pattern trained by the network.

In the past, HONN's were evaluated for performing non-linear discrimination but were rejected as impractical due to the combinatoric explosion of high-order terms. Recent research has shown that the problem of combinatoric explosion can be overcome by building the invariances mentioned above into the neural network architecture using information about the relationships expected between input x_j's.

Now a definition of invariance will be given ([Giles et al., 1988], [Giles et al., 1987], [Reid et al., 1989a], [Reid et al., 1989b]). The output of a unit is invariant with respect to the transformation T on the input pattern if

$$\begin{aligned} T[y_i(W_M(i), p(x))] &= y(W_M(i), T[p(x)]) \\ &= y(W_M(i), p(x)) \end{aligned} \tag{4.2}$$

where

$p(x)$ is a signal function (used as an input pattern) evaluated at position vector x.

$W_M(i)$ represents the entire set of weights associated with unit i.

M is the order of the output unit y_i.

To illustrate the method used, first the weight constraints for one-dimensional translation invariance in a continuous formulation is given. To keep track of the units the continuous variable u is used instead of index i. Eq. (4.1) now becomes:

$$\begin{aligned} y[u; W_M(x), p(x)] &= \Theta(\int dx_1 w_1(u; x_1) p(x_1) \ + \ .. \\ &+ \int .. \int dx_1 .. dx_M w_M(u; x_1 .. x_M) p(x_1) .. p(x_M)) \end{aligned} \tag{4.3}$$

The limits on the integrals are defined by the problem.

Let T now be a translation on the input patterns by $-x_0$, so that

$$T[p(x)] = p(x + x_0) \tag{4.4}$$

where x_0 is the translation of the input pattern. Then, assuming the invariance relation holds, from Eq. (4.2)

$$\begin{aligned} T[y(u; W_M(x), p(x))] &= y[u; W_M(x), p(x + x_0)] \\ &= y[u; W_M(x), p(x)] \end{aligned} \tag{4.5}$$

Since $p(x)$ is arbitrary, term-by-term equality in the argument of the threshold function must be imposed; i.e.

$$\int dx_1 w_1(u; x_1) p(x_1) = \int dx_1 w_1(u; x_1) p(x_1 + x_0) \tag{4.6}$$

$$\begin{gathered} \int \int dx_1 dx_2 w_2(u; x_1, x_2) p(x_1) p(x_2) = \\ \int \int dx_1 dx_2 w_2(u; x_1, x_2) p(x_1 + x_0) p(x_2 + x_0) \\ etc. \end{gathered} \tag{4.7}$$

Making the substitutions:

$x_1 \Rightarrow x_1 - x_0,\ x_2 \Rightarrow x_2 - x_0$, etc.

in the right side parts of Eq. (4.6) and Eq. (4.7), it is found that

$$\int dx_1 w_1(u;x_1)p(x_1) = \int dx_1 w_1(u;x_1 - x_0)p(x_1) \tag{4.8}$$

$$\begin{gathered} \int\int dx_1 dx_2 w_2(u;x_1,x_2)p(x_1)p(x_2) = \\ \int\int dx_1 dx_2 w_2(u;x_1 - x_0, x_2 - x_0)p(x_1)p(x_2) \\ etc. \end{gathered} \tag{4.9}$$

Of course the limits of the integrals on the right side must be adjusted to satisfy the change-of-variables.

In the case of translation invariance, the constraint on the functional form of the weight function requires for the second-order weight to be a function of only vector differences; i.e.

$$\begin{aligned} w_1(u;x_1) &= w_1(u) \\ w_2(u;x_1,x_2) &= w_2(u;x_1 - x_2) \end{aligned} \tag{4.10}$$

In a same way a scale operator S can be defined in such a way that

$$Sp(x) = a^n p(ax) \tag{4.11}$$

where a is a scale factor and a^n is used for normalisation purposes.

In case of the combination of scale and translation invariances, the question arises what will be the effect of the order of the transformations. For certain cases in which the order is unimportant, for second-order weights in the two-dimensional case the following is found:

$$w_2(u;x_1,y_1;x_2,y_2) = w_2(u;(x_1 - x_2)\ /\ (y_1 - y_2)) \tag{4.12}$$

Then w_2 is invariant to scale and translation.

HONN's are thus well suited for invariant pattern recognition because the invariances achieved require no training to produce and apply to any input pattern trained by the network.

[Schmidt et al., 1993] state that HONN's are a form of preprocessing for the standard backpropagation neural networks that use geometrically motivated non-linear combinations of scene pixels to achieve invariant pattern recognition feature spaces. In case of proper choice of HONN pixel combinations they show that it is possible to directly incorporate geometric invariance properties such as translation and rotation invariance into the HONN. In some cases, the hidden layers can be eliminated, and the weight training is achieved by perceptron training.

[Schmidt et al., 1993] were primarily interested in examining the properties of various feature spaces for HONN's, in correlated and uncorrelated noise, such as the effect of various types of input features, feature size and number of feature pixels, and effect of scene size. They also considered HONN training robustness in terms of target detectability. For the experimental set-up they used a 15x20 binary pixel scene possibly containing a 3x10 target. The network used had zero or two hidden layers with up to five neurons per hidden layer. The network training set consisted of 500 scenes (250 with targets and 250 without targets). The network was trained by multiple (100 to 300) presentations of each scene. They indicated that networks with one and two hidden layers gave results equivalent to those with zero hidden layers (with perceptron training). In networks with zero hidden layers the non-linearities are introduced as the network inputs, whereas in networks with one or more hidden layers the non-linearities are learned by the network. The benefit of a network with zero hidden layers is reduced training time. The addition of hidden layers seems to be redundant for targets with simple shapes.

[Reid et al., 1989a] demonstrated a third-order neural network that distinguishes between classes of binary patterns regardless of their translational position, scale, and angular orientation. A significant feature of this network was that it was trained on only one view of each pattern, using a simple single-layer perceptron learning rule! They trained the network to distinguish between the letters T and C at any position, scale or rotation in a 9x9 binary image field, with 100% accuracy in a noise-free background. For their tests they used

1. a second-order neural network to achieve simultaneous invariance to translation and scale;

2. a third-order neural network to achieve simultaneous invariance to translation, scale and in-plane rotation.

The second-order network they simulated, used a 16x16 (or 256 nodes) input window fully interconnected to a single output node. They trained the network on just two distinct patterns - only one size and one location for each pattern. It learned to distinguish between the patterns after approximately ten passes of the training set. It successfully distinguished between all translated and scaled versions of the two objects with 100% accuracy. With this network they also trained the T-C problem.

For the third-order network simulation they used an input window of 9x9 pixels, or 81 nodes. After approximately 20 passes through the training set, the network learned to distinguish between distortions of two objects with 100% accuracy. The

T-C problem could be trained with full invariance to translation within the input field, to scale over a factor three, and to 90^o rotations.

[Spirkovska et al., 1989] state that because of the memory required for the large number of interconnections in a fully connected HONN, the size of the input field is very limited. To achieve, for example, invariance to translation, scale, and in-plane rotation a third-order network can be used. For a 9x9 pixel input field 85,320 interconnections are formed, whereas increasing the resolution to 128x128 pixels increases the number of interconnections to $7.3 * 10^{11}$.

They describe four different strategies for non-fully connected HONN's that make it possible to use input fields of bigger sizes. They have shown that by using such strategies an input field of 128x128 pixels can be attained while still achieving in-plane rotation and translation invariant recognition. The four strategies used are:

1. Local connectivity
   The assumption behind local connectivity is that local connections are adequate for distinguishing between objects and that distant information will be represented by some combination of local information.

   Local connectivity

2. Sampled connectivity
   In sampled connectivity, only a fraction of all possible input pixel triplets are connected to the output. This fraction is not a function of the distance between pixels. Only a predetermined subset of interconnections (for example, every ith one) is used.

   Sampled connectivity

3. Probabilistic connectivity
   Probabilistic connectivity attemps to improve on local connectivity. The images are sampled using a technique in which the probability of connections varies with distance.

   Probabilistic connectivity

4. Regional connectivity
   Regional connectivity combines the advantages associated with the other approaches while eliminating their disadvantages. Triplets of pixels are connected to the output only if the distance between all of the pixels falls within a set of preselected regions. Triplets are formed from pixels which are local, distant or in-between. Further, because the connections are deterministic, it is guaranteed that the same information will be sampled from a rotated or a translated test image as from the training image. Additionally, since pixel triplets of various separations are sampled, some amount of scale invariance can be expected.

   Regional connectivity

[Kanaoka et al., 1992] show that scale invariance does not hold precisely for small input space and that HONN's are not so effective for distortion of unknown patterns. They state that in high-order terms, the third-order is the minimum order to achieve the invariance of shift (translation), scale and rotation of patterns. The general third-order formula is:

$$y_i \;=\; \Theta(\textstyle\sum_j \sum_k \sum_l w_{ijkl}\, p(x_j) p(x_k) p(x_l))$$

Scale invariance is not achieved for a pattern larger than the training one. For these patterns there exist triangle cells which did not fire at the training, and in a training pattern there are triangle cells which did not fire at the discrimination for a smaller unknown pattern (figure 4.4). Therefore, they came with the idea to represent the input patterns as feature points and the connections among them. As shown in figure 4.5, A connects to B directly but A' does not connect to B'. Thus triangle cell $A'B'C'$ can be treated as different from triangle cell ABC.

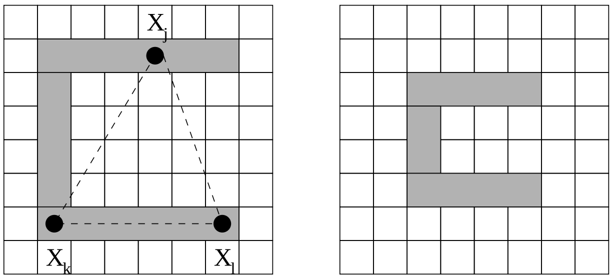

Figure 4.4:

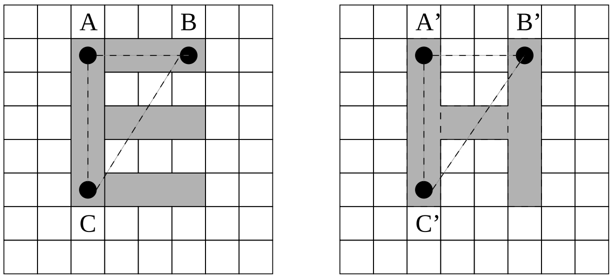

Figure 4.5:

As an approach to distorted patterns they introduced the following similarity measure between triangle cells. Let ABC be a fired triangle cell at the training, and let $A'B'C'$ be a fired triangle cell for an unknown pattern and not belonging to the set of cells weighted at the training. The similarity measure between ABC and $A'B'C'$ is then be defined as follows:

$$S_{A'B'C'}(ABC) = (|\alpha - \alpha'| + |\beta - \beta'| + |\gamma - \gamma'|) / \pi$$

$$\alpha < \beta < \gamma \text{ and } \alpha' < \beta' < \gamma'$$

where α, β and γ are the respective angles of triangle ABC and α', β' and γ' are the respective angles of triangle $A'B'C'$.

ABC and $A'B'C'$ further have the same connectivity information. Furthermore,

$$S_{A'B'C'}(A^*B^*C^*) = min\{S_{A'B'C'}(ABC) \mid S_{A'B'C'}(ABC) \leq \varepsilon, ABC \in T\}$$

where T is the set of triangle cells weighted at the training. Thus triangle cells $A'B'C'$ is the one most similar to $A^*B^*C^*$, and $A^*B^*C^*$ is regarded to be fired on behalf of $A'B'C'$.

For validation of their method they considered the simple problem of distinguishing between the characters E and H on a 11x11 pixel network. Feature points were extracted by using eight-neighbour connectivity. Using feature points, connectivity and the similarity measure in two iterations on only two input patterns (one for the E and one for the H) they got good recognition accuracy. With increasing ε to 4 the recognition accuracy reached 100%.

[Linhart et al., 1992] used a HONN combined with first-order networks and the mechanisms known from Adaptive Resonance Theory (ART) to implement position invariance recognition efficiently. Its recognition is based on a 16x16 pixel input which contains a section of the image found by a separate centring mechanism. Figure 4.6 schematically depicts the overall architecture.

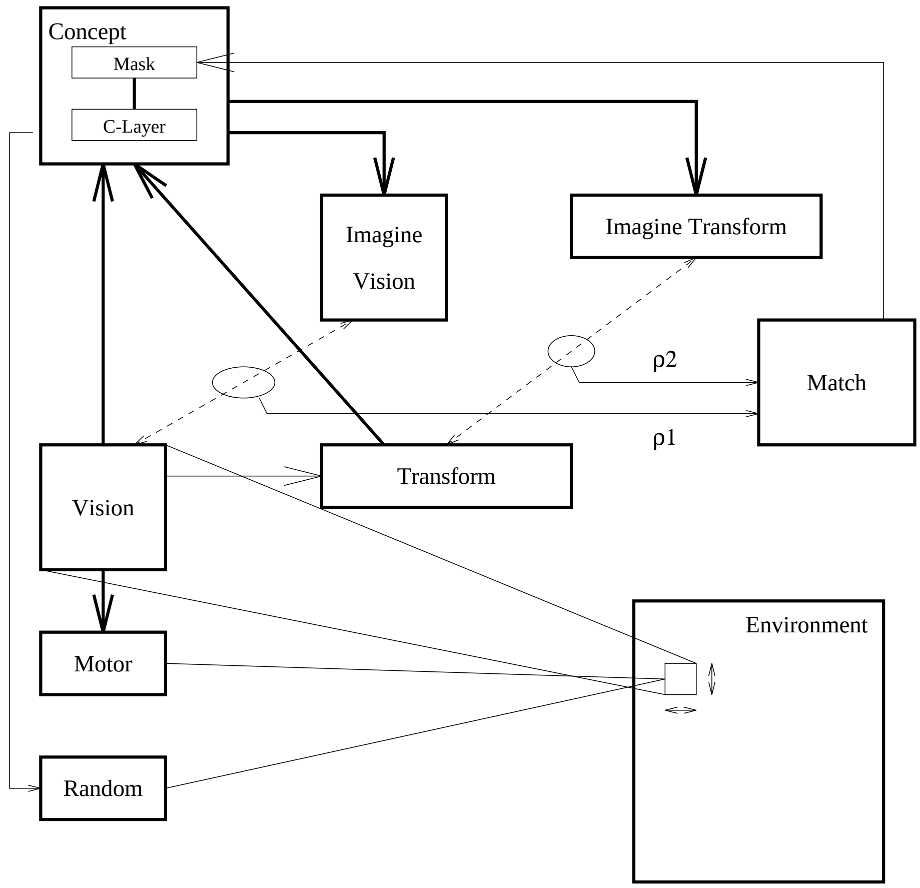

Figure 4.6: Architecture of the autonomous learning model (schematically)

The input layer (*Vision*) contains the current section of the larger image (*Environment*). The exploration component choosing a particular section is controlled by a motor layer (*Motor*), which consists of two trinary units corresponding to two directions of movement (X and Y), and a random component (*Random*). In addition - whenever necessary - a pattern in the transform layer (*Transform*) on the basis of the size-invariant second-order neural network is activated. Categorization is done in the C-layer which, together with a masking layer (*Mask*), forms the conceptualization component. Two additional layers - *Imagine Vision* and *Imagine Transform* - are introduced for implementing class prototypes and the comparison with the corresponding vision and transform layers.

The overall system essentially consists of a modified ART model. In addition to a feedback loop for resonance of first-order categories in the input, it contains a similar second-order loop. Furthermore, an exploration component is added which automatically and sequentially generates patterns to be learned. This component consists of a centring mechanism which focusses on a 16x16 pixel part of the 100x100 pixel input image.

This model, training autonomously, combines a variety of advantages of the different network approaches employed. The first-order network permits all categorization tasks inherent to the ART model, whenever the position of the pattern in the input layer is fixed. In addition, the second-order network permits the model to recognize trained patterns also on arbitrary different positions in the input.

A cycle of recognition starts with the update from the vision to the C-layer. The winner of the C-layer is selected (through Winner-Takes-All) and activation propagated to the imagine vision layer. For known patterns a prototype is activated, which is compared to the pattern in the vision layer. If the similarity between the patterns is above the value of vigilance parameter ρ_1, the category is accepted. If it is not, the category is not immediately rejected (as in regular ART). Via the size-invariant second-order neural net a pattern in the transformation layer is activated and compared to the pattern in the imagine transform layer, after propagation via the C-layer. A second vigilance parameter ρ_2 decides upon acceptance or rejection of the category. In this way, different patterns on the same position are categorized through the first-order network, and similar patterns of different positions through the size-invariant second-order neural net.

They tested this model with several handwritten digits put together on a 100x100 pixel input image. Each digit was included several times in different shapes. Depending on the vigilance parameter 100% correct recognition was achieved. In the case of two shapes of the same digits that were too dissimilar, however, misclassifications could be observed. For all tests, position invariance was perfect, size and distortion invariance only limited.

4.4 Classification

After preprocessing of the raw image data classification methods can be used to separate data following relations present in the data. One of the classification methods used is the one in which neural networks are used.

During the training phase a neural network has learned to distinguish between relations in the training set. During the classification phase new, unknown image data are presented to the neural network. The neural network tries to find one or more of the relations learned in this image. If it succeeds in finding a relation the neural network classifies the image as belonging to the class of images with this relation. If more than one of the relations are found this can lead to an erroneous classification. This may happen due to the fact that the training set offered to the network was not complete, i.e. in the images in the training set not enough distinction between the relations was present, or due to the fact that the network had not trained enough, i.e. had not learned to distinguish clearly between all relations.

[Lippmann, 1989] gave a division into four groups of neural network classifiers used in pattern classification. These groups are:

1. Probabilistic classifiers

2. Hyperplane classifiers

3. Receptive field or kernel classifiers

4. Exemplar classifiers

He based this division on practical differences between classifiers and internal differences in how classifiers form decision regions. Furthermore, [Lippmann, 1989] gives a survey of neural networks used in pattern recognition and the fields of research they are used in. These are:

- Backpropagation
  - Classify speechs sounds
  - Form text-to-phoneme rules
  - Discriminate between underwater sonar echoes.
  - Perform non-linear signal processing
- Decision tree classifiers
  - Pattern classification
  - Artificial intelligent applications
- Higher-Order Neural Networks (HONN)
- Feature-Map classifiers
- Learning vector quantizer
- Hypersphere classifiers
  - Complex decision problems
- Radial Basis Function classifiers
  - Vowel and digit speech recognition

After using texture and physical parameters for preprocessing their image data [Gebbinck et al., 1993] used three different methods to determine the ability to detect and classify various categories of diffuse liver disease. The three methods used are:

1. Discriminant analysis
2. Backpropagation
3. A self-organizing feature map with the conscience mechanism as training procedure.

Their results show that for the four disease classes they used to discriminate between, backpropagation always gives the best results with a specificity ranging from 86.8% to 99.2%, and a sensitivity ranging from 81.8% to 100.0% and correct classification ranging from 84.7% to 99.4%. Discriminant analysis presented the worst results. The disadvantage of backpropagation was the time needed to calibrate the classifier, while discriminant analysis had the fastest time for calibration.

[Pan et al., 1992] used self-organizing mapping algorithms for liver tissue classification. They compared the Dynamic Neuron Adding (DNA) method with the two Learning Vector Quantization methods (LVQ1 and LVQ2). With respect to the training time, accuracy rate or stability, they found DNA to be always better than both LVQ methods. Furthermore, the DNA method requires substantially fewer neurons. For LVQ1 they found an accuracy rate for training data between 84.0% and 88.0%, for LVQ2 this rate was between 87.6% and 93.3% and finally for DNA this rate was between 89.3% and 93.3%.

4.5 Discussion

In the previous sections the authors mentioned used different kinds of neural networks for very different kinds of jobs. They stated advantages and disadvantages of the networks used and they also compared results of different networks for the same job.

Although the jobs on which the networks were tested are almost always much simpler than the job the network will be used for in this study, the results these authors found can be easily extrapolated as the advantages and disadvantages they found will be the same for this work.

Results found by the authors state that first-order networks produce solutions that are specific to particular training sets whereas this fact does not hold for HONN's. Furthermore, the architectural complexity of HONN's is low compared to networks using first-order units.

For backpropagation they found that with a large number of training passes and an exhaustive training set only a recognition accuracy between 80% and 90% can be reached. HONN's have an effective and rapid training. Only up to about 20 presentations of 1 example of each training pattern are needed to reach a recognition accuracy up to 100%. Furthermore, because of the low number of presentations and the small training set the training speed of a HONN is superior to that of backpropagation. Backpropagation can also get stuck in a local minimum, thus not training and classifying correctly.

Another advantage of HONN's is that no training is required to achieve invariances as they can be built into the architecture of the network. First-order neural networks have to learn the invariances (with a chance of incorrect learning) which causes the tremendous increase of presentations of patterns and training sets.

Furthermore, HONN's, as opposite to first-order neural networks, use the relationships between pixels. Therefore, they can most probably be used to learn images.

Last, HONN's have since long been known in literature and many researchers have already used them in research.

HONN's also have some disadvantages, as they have difficulties with scale invariances and distorted patterns. Solutions for these problems are, however, available ([Kanaoka et al., 1992]).

Comparing the different arguments the advantages of HONN's far outweigh the disadvantages and also far outweigh the advantages of first-order neural networks for the kind of images the neural network will now be needed for. Therefore, in the rest of this thesis the HONN approach is chosen.

Chapter 5

Pilot study

5.1 Introduction to the pilot study

In this chapter a study is carried out to investigate the usefulness of the HONN approach. This study and its results are described in the next sections. For the neural network part the control and visualization environment ConVis ([Vuurpijl, 1994]) is used, which package provides control structures for the use of neural networks and tools for the visualization of the results produced. The co-occurrence parameters are calculated using the simulation package Simplan. Finally, Matlab is used to compare the results of the neural networks and the co-occurrence parameters. Using this package these results can be nicely visualized.

Framework of the pilot study
The HONN first has to show its usefulness with one dimensional, simulated signals. As they are rather simple the HONN will have to be very accurate. Only then it might be useful for more complex signals. The way these signals are generated will be discussed in section 5.2.

An approach will be used for the HONN which is different form the one described in literature (section 4.3). Instead of building the invariances mentioned in this section into the weights of the connections between the neurons, they will be fed to the input neurons of a First-Order Neural Network (FONN) by using the autocovariance function (figure 5.1). The validation of this approach will be proved in section 5.3.

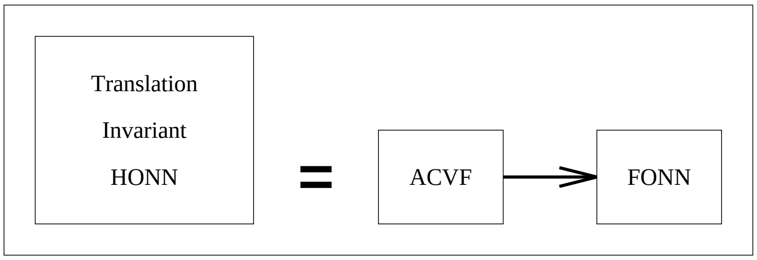

Figure 5.1: HONN approach in this section

For the FONN, beside the use of the perceptron network as described in literature, also the LVQ and the backpropagation networks will be used. Implementations of these neural networks will be described in section 5.4.

In section 5.5 the conditions under which the co-occurrence parameters are calculated, will be mentioned.

In section 5.6 test results of the three HONN approaches will be provided.

Next, in section 5.7 the results of the co-occurrence matrix method will be compared with the results of the HONN approach. From the co-occurrence matrix the following parameters will be calculated for this comparison (section 3.2):

1. Contrast
2. Angular second moment
3. Entropy
4. Correlation

These statistical parameters will be employed as they are the ones most often used for tissue characterization.

Before comparison results are given, first a measure for comparing the results of the HONN approach with the co-occurrence matrix method will be explained.

In section 5.8 conclusions about the usefulness of the HONN approach compared with the co-occurrence matrix method will be given.

5.2 Generating signals

For the simplest test of the usefulness of the HONN approach one dimensional, generated signals are used, which must have the following properties:

1. Two different signals only differ in their texture. This means that only the second-order, spatial statistics of the underlying, generating model are changed. Point average and variance remain equal.
2. Texture is characterized by:
   - correlation distance (equivalent to the FWHM)
   - structure distance (equivalent to the grid distance)
   - relative structure amplitude (equivalent to the relative, structural scatterer strength)

The signals are roughly generated as follows; first a Gauss filter is created, then the convolution sum of this filter with random, uncorrelated noise is calculated, and from the resulting signal the first 128 samples are taken, which are supposed to contain all the necessary information. Finally, the autocovariance of the 128 samples is calculated. From the final autocovariance signal only the first 64 samples are taken for reasons mentioned in subsection 5.2.1.

In the next paragraphs each of these steps will be described in more detail.

The Gauss filter

Gauss kernel The Gauss kernel has the same shape as the normal probability density function (Gaussian or normal distribution):

$$h(t) = \frac{1}{\sigma\sqrt{2\pi}} * \exp -((t - \mu)^2 / 2\sigma^2)$$

where

σ is the standard deviation of $h(t)$.
μ is the average or mean value of $h(t)$.

The Gauss filter can be based either on one kernel (figure 5.2) or on a summation of displaced kernels (figure 5.3), where the amplitude of the displaced kernel is multiplied by a decreasing factor with an initial value smaller than 1. This multiplication makes sure that most of the energy will be around the central part of the filter.

$$h_{disp.}(t) \;=\; A\,\frac{1}{\sigma\sqrt{2\pi}} \;*\; \exp -((t \;-\; \mu_{disp.})^2 \;/\; 2\sigma^2), \qquad A \;<\; 1$$

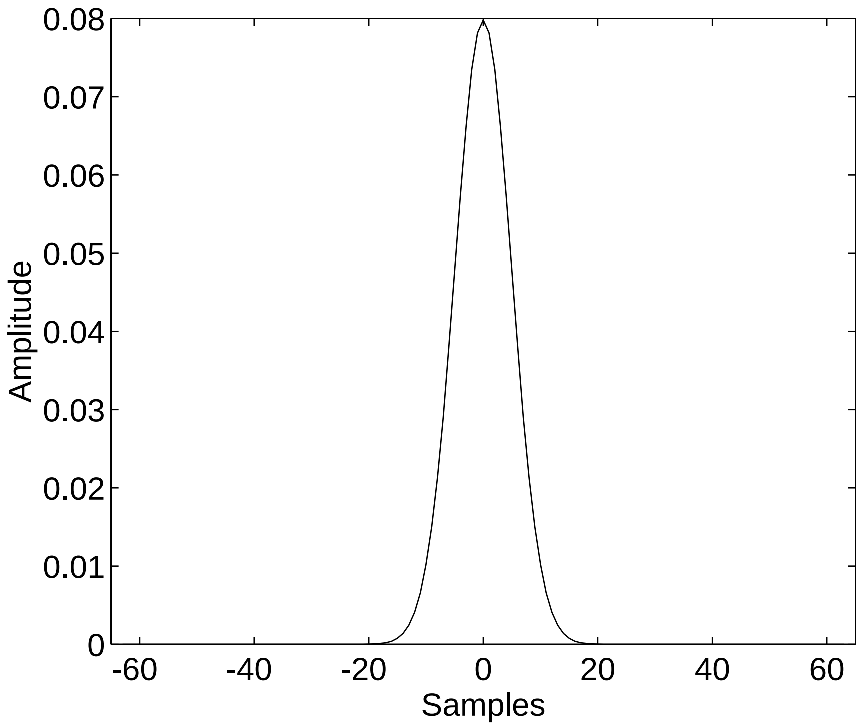

Figure 5.2: Gauss filter based on one kernel: Filter length 127, Sigma 5

When displaced kernels are used it has to be made sure that the filter length is chosen long enough. Otherwise, unwanted effects are introduced in the tails of the final filter due to the fact that some of the displaced kernels are abruptly cut off.

The Gauss filter created so far is finally to be smoothed with a Hanning window (figure 5.4) which follows the formula:

Hanning window

$$g(t) \;=\; h(t) \;*\; w(t)$$

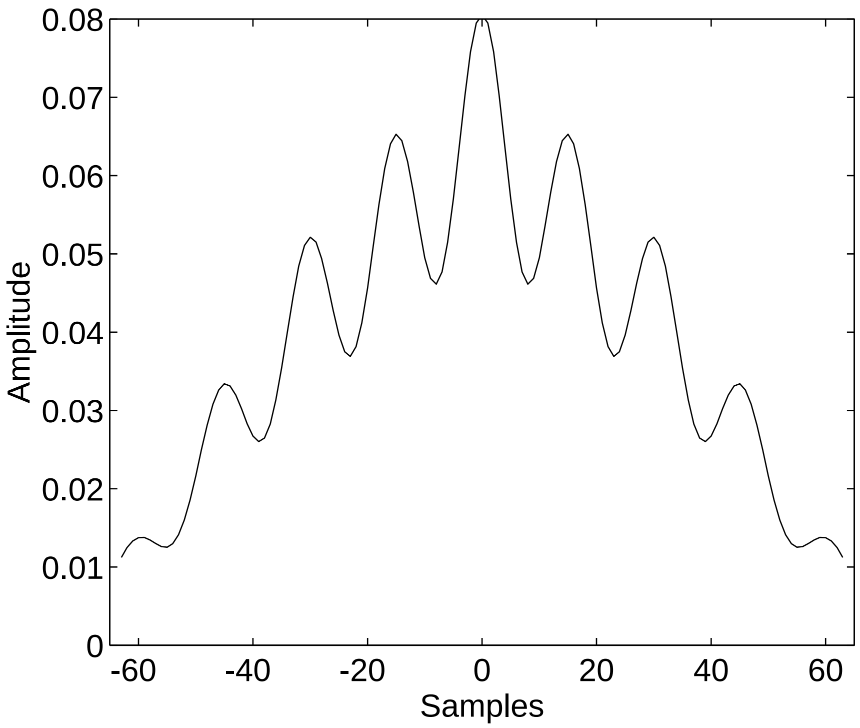

Figure 5.3: Five kernels as a base of a Gauss filter, after summation: Central one with Filter length 127, Sigma 4, Displaced ones with Filter length 127, Sigma 4, Mean 15, Multiplication factor 0.8

where $w(t)$ is:

$$w(t) \;=\; \frac{1}{2}\,[1 \;-\; \cos 2\pi t]$$

The use of this Hanning window is necessary to smooth the Gauss filter when it is abruptly cut off due to too small a filter length (figure 5.3).

The result after smoothing with the Hanning window is the Gauss filter which is used in this study.

The test signal

Test signal

The test signal is generated using the following formula:

$$y(t) \;=\; g(t) \;\otimes\; x(t)$$

where $x(t)$ is uncorrelated, uniformly distributed noise in the interval [-0.5, 0.5] and $\otimes$ means that the convolution of $g(t)$ and $x(t)$ is calculated.

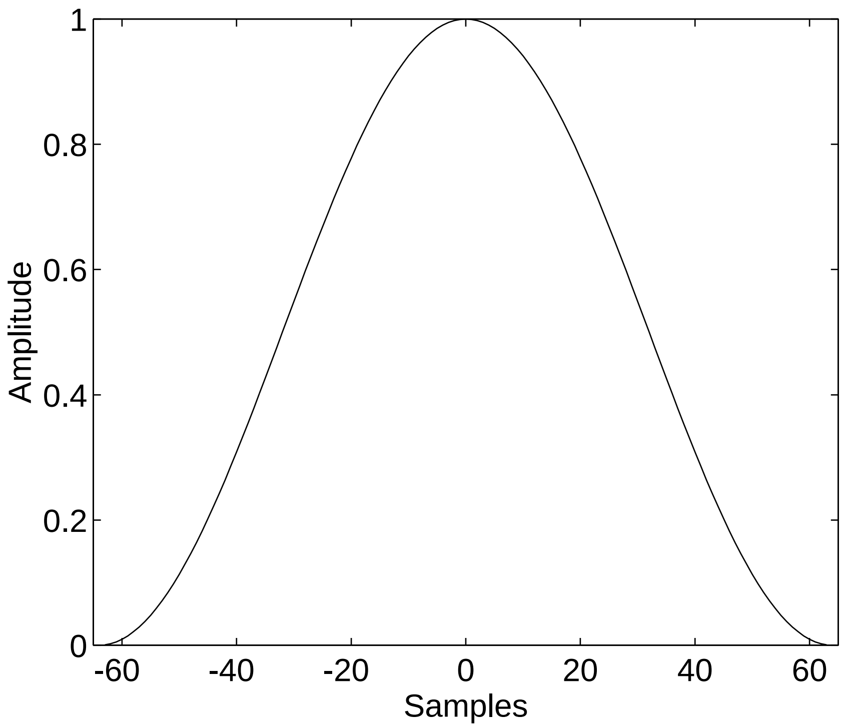

Figure 5.4: Hanning window: Number of samples 127

The autocovariance

Autocovariance

In practice the mean and variance of the uniformly distributed noise x cannot be precisely determined since an exact knowledge of the PDF of x will not generally be available. Hence one must be content with estimates of the mean value and the variance based upon a finite number of observed values (in this study 128).

Estimation can be biased and unbiased. If

$$E\left[\hat{\phi}\right] = \phi$$

holds, in which $\hat{\phi}$ is an estimator for the parameter ϕ, then the estimator is said to be unbiased; otherwise it is said to be biased.

Using this, the biased and unbiased autocovariances are implemented as follows ([Oppenheim et al., 1975]):

$$\begin{array}{lll} \text{Biased:} & \hat{\gamma}_{xx}(m) = \frac{1}{N}\sum_{i=1}^{N-|m|-1} x_i x_{i+m} & \\ & & m = 0, \pm 1, \cdots, \pm(N-1) \\ \text{Unbiased:} & \hat{\gamma}_{xx}(m) = \frac{1}{N-|m|}\sum_{i=1}^{N-|m|-1} x_i x_{i+m} & \end{array}$$

After dividing the autocovariance by $\bar{\gamma}_{xx}(0)$ (normalization) we have the correlation coefficient.

Both the normalized, biased and unbiased autovariance estimators have been used on the convolution signals mentioned before. As the autocovariance function is symmetric around zero, only the right part of the autocovariance has been taken. Due to symmetry this can be done without discarding information. The results for one signal (128 samples) can be seen in figure 5.5.

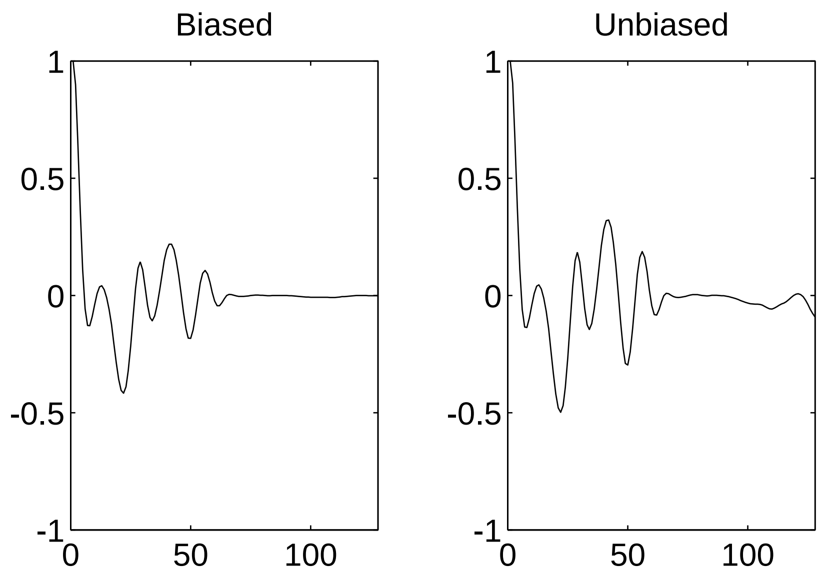

Figure 5.5: Correlation coefficient of 1 noise signal filtered with a Gauss filter

Furthermore, the normalized, biased and unbiased estimations of autocovariances of ten signals have been averaged to reduce random noise (figure 5.6).

5.2.1 Using the autocovariance as HONN input

Looking at figures 5.5 and 5.6 it can be seen that the estimation of the autocorrelation causes some problems for greater lags:

- For greater lags it is difficult to estimate the correlation due to the small window size.
- The unbiased estimate of the correlation yields much noise in the tail, which is suppressed in the biased estimate.

The noise in the tail of the unbiased estimate contains hardly any information, but suppresses the real information. The biased estimate suppresses most of the noise, but also part of the real information.

Therefore, to get around this problem it has been decided only to take the samples in the first half of the window size of the autocovariance as input data for the HONN. This means that in the following either biased or unbiased autocovariances can be used. For the rest of this study the biased version will be used.

Although in the original version of the HONN the autocorrelation is used, in this thesis the autocovariance will be used. This can be done because the product of the means in the autocorrelation is (almost) equal to zero. This makes the autocorrelation equal to the autocovariance which can thus be used as input for the HONN.

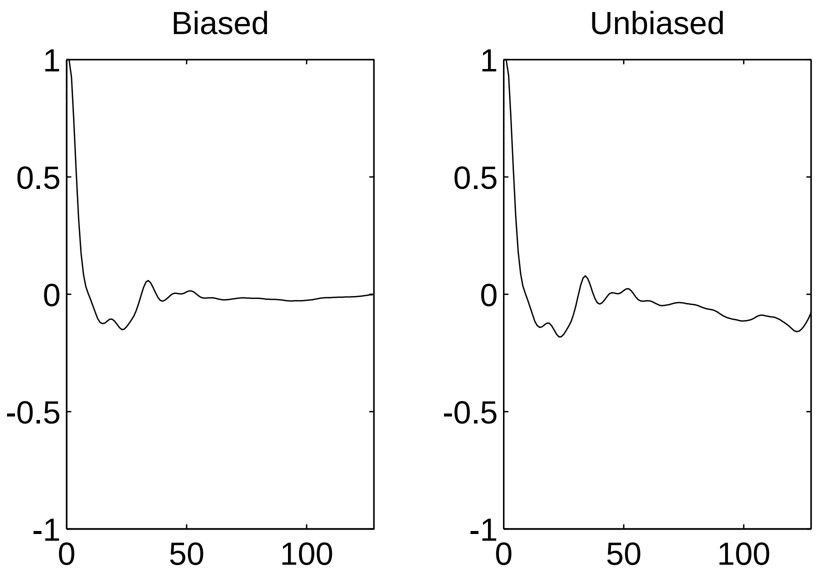

Figure 5.6: Average correlation coefficient of 10 correlation coefficients of noise signals filtered with a Gauss filter

5.2.2 Using the test signal to calculate co-occurrence parameters

From the test signals generated for each class, windows will be formed. For each of these windows the co-occurrence matrix is calculated. This matrix is then used to calculate the four co-occurrence parameters mentioned in section 3.2:

- Contrast
- Angular Second Moment
- Entropy
- Correlation

5.3 The translation invariance constraint

A simple Higher-Order logic unit is described by ([Giles et al., 1987]) (notation as used in section 4.3):

$$\begin{aligned} y(p(x_p)) &= S\,[\,net(p(x_p))] \\ &= S\,[\,T_0(p(x_p)) + T_1(p(x_p)) + \cdots + T_k(p(x_p))] \end{aligned} \tag{5.1}$$

where

$y(p(x_p))$	is the output of the ith high-order neuron unit with input $p(x_p)$.
S	is a sigmoid function.
$T_n(p(x_p))$	is the nth order term for the ith unit with input $p(x_p)$.
K	is the order of the unit.

When the order of the units does not exceed 1, a normal neural network like backpropagation or hopfield is dealt with.

The second-order term is a linear weighted sum over second-order products of inputs:

$$T_2(p(x_p)) \quad = \quad \sum_j \sum_k w_{ijk} p(x_p)_j p(x_p)_k \tag{5.2}$$

The second-order perceptron update rule can be expressed as

$$w'_{ijk} \quad = \quad w_{ijk} \; + \; [t_i(p(x_p)) \; - \; y_i(p(x_p))] p(x_p)_j p(x_p)_k \tag{5.3}$$

where

$t_i(p(x_p))$ is the target output of the ith unit with input $p(x_p)$.

A unit is invariant under the action of a transformation group G = {g} *iff*:

$$y_i(gp(x_p)) \quad = \quad y_i(p(x_p)) \quad \forall g \in G \tag{5.4}$$

This invariance is imposed by averaging the weight matrices over the group, thus eliminating the unit's ability to detect correlations which are incompatible with the imposed group invariance. In particular a G-invariant unit can be handcrafted from the higher-order logic unit described by Eq. (5.1). This handcrafted unit is described by the equation

$$y_i(p(x_p)) \quad = \quad S\left[\sum_{g \, \in \, G} net(gp(x_p))\right] \tag{5.5}$$

where the sum is over all members of the group G and the *net* operator is defined as in Eq. (5.1). To see that this unit is invariant under the group G, note that for all $h \; \in \; G$, the following applies

$$\begin{aligned} y_i(hp(x_p)) \quad &= \quad S\left[\sum_{g \, \in \, G} net(ghp(x_p))\right] \\ &= \quad S\left[\sum_{g' \, \in \, G} net(g'p(x_p))\right] \\ &= \quad y_i(p(x_p)) \end{aligned} \tag{5.6}$$

where the substitution $g' = gh$ has been made. The second step follows from the fact that a sum of $g = g'h^{-1}$ is equivalent to a sum over g', since multiplying all terms in a sum over a group by a member of the same group simply results in a permutation of the terms in the sum.

Suppose a second-order unit is given in which translation invariance is to be implemented. This unit is described by

$$y_i(p(x_p)) = S\left[\sum_j \sum_k w_{ijk} p(x_p)_j p(x_p)_k + \cdots\right] \tag{5.7}$$

It is to be determined what constraints must be placed on the w matrix in order to ensure that the unit's output y is invariant under the transformation group T. Define $T = \{g(m)\}$ such that $g(m)p(x_p)_k = p(x_p)_{m+k}$ (m is the translation factor). If this group operator is applied to the previous equation the result is

$$y_i(p(x_p)) = y_i(g(m)p(x_p)) \quad \forall g(m) \in T \tag{5.8}$$

Expanding the right-hand side of this equation yields

$$\begin{aligned} y_i(g(m)p(x_p)) &= S\left[\sum_j \sum_k w_{ijk} p(x_p)_{j+m} p(x_p)_{k+m} + \cdots\right] \\ &= S\left[\sum_j \sum_k w_{i(j-m)(k-m)} p(x_p)_j p(x_p)_k + \cdots\right] \end{aligned} \tag{5.9}$$

where in the last step it is assumed either that periodic boundary conditions are appropriate or that edge effects are negligible. A comparison of the above equations reveals that the constraint that must be placed on w_{ijk} to ensure translation invariance is

$$w_{ijk} = w_{i(j-m)(k-m)} \tag{5.10}$$

Here w_{ijk} is a function of a pattern composed of two points located at j and k. This constraint stipulates that w_{ijk} has the same value when evaluated at any pattern $(j-m, k-m)$ which is a translated version of the original pattern (j,k).

Another way of expressing the invariance constraint come to in the previous paragraph becomes clear if it is noted that the set of translation equivalence classes of two points is given by the relative coordinate $dj = j - k$. In other words, every possible value of dj corresponds to a single equivalence class and vice versa. It is now obvious that this condition is equivalent to constraining w_{ijk} so that it depends only on the relative coordinate dj and not on the absolute coordinates j and k. This leads to the condition on w_{ijk} that

$$w_{ijk} \quad \rightarrow \quad w_{idj} \tag{5.11}$$

Thus, the implementation of translation invariance in Eq. (5.1) is equivalent to redefining the correlation matrices so that they depend only on relative position dj and not on absolute position j. The second-order invariant term is

$$T_2(p(x_p)) \quad = \quad \sum_{dj} w_{i\,dj} \sum_{j} p(x_p)_j p(x_p)_{j\,+\,dj} \tag{5.12}$$

Here, the scale factor w_{ijk} is learned by the neural network and w_{ijk} weights the autocorrelation.

An error-correcting perceptron learning rule for the second-order term can be written as

$$w'_{i\,dj} \quad = \quad w_{i\,dj} \;+\; [t_i(p(x_p)) \;-\; y_i(p(x_p))] \sum_{j} p(x_p)_j p(x_p)_{j\,+\,dj} \tag{5.13}$$

Now reference is again made to the first-order, error-correcting, perceptron learning rule looking like:

$$w'_{ij} \quad = \quad w_{ij} \;+\; [t_i(p(x_p)) \;-\; y_i(p(x_p))] p(x_p)_j \tag{5.14}$$

where $p(x_p)$ is the input to the neural network.

Comparing Eq. (5.14) with Eq. (5.13) it can easily be seen that a HONN can especially be built in two ways:

1. The autocorrelation is built into the learning rule of the network (Eq. (5.13)) and the network is fed with raw data. In essence this provides a neural network with second-order units.

2. The autocorrelation is not built into the learning rule of the network (Eq. (5.14)), but the autocorrelation is calculated first and its result is fed to the neural network. In essence this results in a neural network with first-order units.

As the two ways of building a HONN are proven to be equal and the second option is the easiest one to build, in the rest of this chapter the second way of building a HONN is used.

5.4 Implementation of the neural networks

From the implemented perceptron, backpropagation and LVQ networks, the standard learning rules for LVQ as can be found in [Kohonen, 1984], [Hertz et., 1991] and [Rutten et al., 1992] and perceptron (generalized delta rule). For backpropagation the generalized delta rule is extended with *momentum* to remember previous weight changes, and *psi* to determine the steepness of the activation function. These networks are implemented in C and are used in combination with ConVis. The use of ConVis makes the setting of parameters, variables etc. flexible and easy and also allows for easy visualization of results.

For the initialization part of LVQ a couple of algorithms can be followed. The choice is made to use the algorithm below ([Rutten et al., 1992]):

```
For all classes 'C' {
    determine the average vector V for this class;
    divide the number of neurons into an equal number per class;
    allocate this number of neurons with weight vectors
    initialized somewhere around V;
}
```

As there is no interest in classification numbers of the input patterns (the standard output of the implementation of LVQ, backpropagation and perceptron used), but in the real values of the output neurons, an option was implemented in each of these neural networks to deliver these real values. In case of backpropagation and perceptron this means that the activations of the output neurons are now delivered. In case of LVQ the euclidean distances of the output neurons and their weight vectors are now delivered.

5.5 Calculation of the co-occurrence parameters

For the test sessions in this chapter co-occurrence parameters have to be calculated from the test signals. Therefore, the 8190 test signals per class are separated into 819 windows of size 128x10. From each of these windows the co-occurrence matrix is calculated, which is then used to calculate the co-occurrence parameters.

For all test sessions in this chapter, the distance used in the calculation of the co-occurrence matrix is arbitrarily set to 5 (equal to the sigma of the autocovariances used in the tests of the HONN). The number of grey levels (256) is reduced to 16. There are two reasons to do reduce the number of grey levels:

1. Speed of calculation. With 256 grey levels the co-occurrence matrix has a size of 256x256 entries, while with 16 grey levels this size is only 16x16 entries. This makes calculations much faster.

2. More important is that, statistically seen, nothing can be said about the values in a 256x256 entries co-occurrence matrix. These values present the number of times a certain grey level can be found next to another grey level. With 256 grey levels the matrix entries only contain zeros and ones. After a reduction of the number of grey levels to 16, chances are high that a certain grey level can be more often found next to another grey level, so the values in the entries will be higher and will thus be of more statistical interest.

5.6 Test results of the HONN approach

To make the test results, which will be obtained in this section, statistically reliable they have to comply with the rule that the ratio of the sample size per class to the number of features has to be greater than three ([Foley, 1972]). As the maximum network size is not known at this moment, but will most probably not exceed 200 neurons, it has been chosen to generate 819 training and 819 test signals (using the steps described in section 5.2) for each class mentioned in the rest of this section.

For parameter initialization of the neural networks values, known from literature, are used. For all the networks used several training sessions were run, varying the parameters slightly, to find the optimal network.

In the rest of this section 'average percentage of correctly classified patterns' means, the average of the values at the diagonal of the confusion matrix of the network. A typical combination of training and test confusion matrices is shown below.

training pattern confusion matrix						
T — O	0	1	2	n	match	fail
0	**99.8%**	0.2%	0.0%	819	817	2
1	0.0%	**67.4%**	32.6%	819	552	267
2	0.0%	20.9%	**79.1%**	819	648	171

test pattern confusion matrix						
T — O	0	1	2	n	match	fail
0	**99.8%**	0.2%	0.0%	819	817	2
1	0.0%	**63.0%**	37.0%	819	516	303
2	0.0%	23.4%	**76.6%**	819	627	192

5.6.1 Results of session 1

Three classes of signals are generated giving a total of 2457 signals for both training and testing. The number of features of the neural networks used will be 64.

For this test the classes of signals have been generated using three different Gauss filters:

Class	**Sigma**	**Mean**	**No of kernels**	**Multiplication factor**
0	5	0	1	not applicable
1	5	10	21	0.5
2	5	10	11	0.6

The HONN approach using LVQ
The 2457 training signals were, in a round robin manner (i.e. following the order in which they were originally presented to the network), fed to the LVQ network. For training the settings of LVQ were initialized as follows:

Learning rate	0.01
Learning rate change	0.99

The network was trained for different numbers of output neurons. Depending on this number of output neurons training took 19 to 229 epochs. After each training session the 2457 test signals were fed to the network. Figure 5.7 shows the average percentage of correctly classified training and test signals respectively.

It has to be said that for signal class 0 training and test correct classification percentages were between 99.8% and 100.0% for all numbers of output neurons, meaning that this signal class could be, almost strictly, separated from the other two signal classes. Signal classes 1 and 2 are partly overlapping, thus giving lower percentages of correct classification.

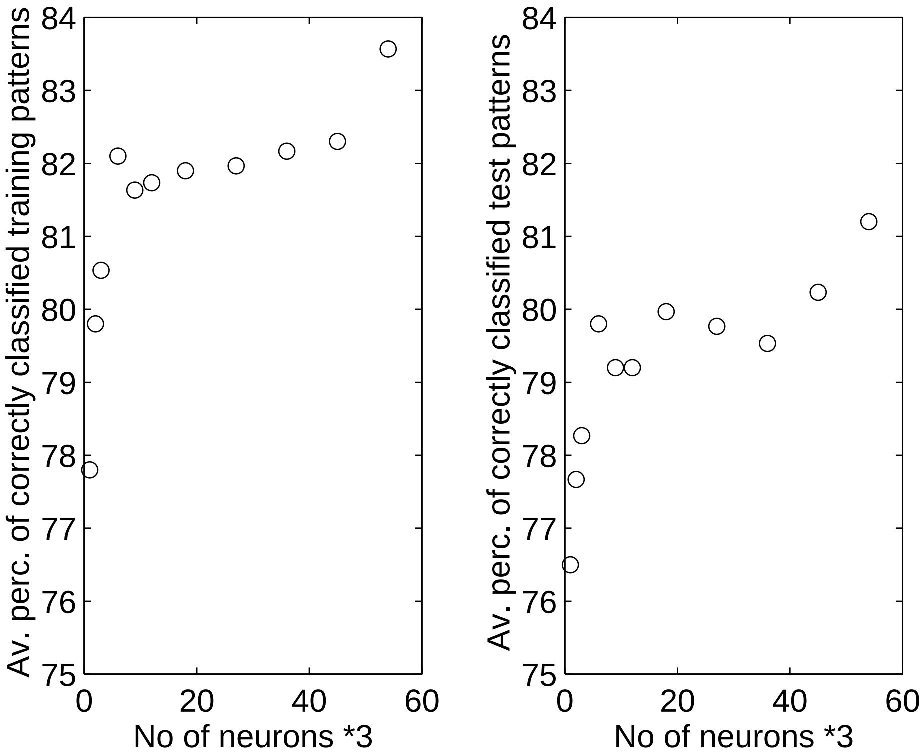

Figure 5.7: Average percentages of correct classifications for training and test using different numbers of output neurons (LVQ)

The HONN approach using Perceptron

The 2457 training signals were, in a round robin manner, fed to the perceptron network. For training the settings of perceptron were initialized as follows:

No of output neurons	3	6	9
Learning rate	0.0001	0.001	0.005

The learning rate had to be varied at the same time varying the number of output neurons as keeping the learning rate equal led to highly fluctuating classes and a fluctuating, not converging, error.

The network was trained for these different numbers of output neurons. Depending on the number of output neurons training took 179 to 458 epochs. After each training session the 2457 test signals were fed to the network. Figure 5.8 shows the average percentage of correctly classified training and test signals respectively.

Also with this test, class 0 training and test correct classification percentages were 100.0% for each configuration while classes 1 and 2 gave training percentages between 64.2% and 81.4% and between 59.7% and 78.5% respectively. For recall they presented percentages between 61.9% and 78.9% and between 57.0% and 76.7% respectively. Furthermore it has to be said that when class 1 showed a high percentage of correct classification, class 2 gave a low percentage of classification and vice versa.

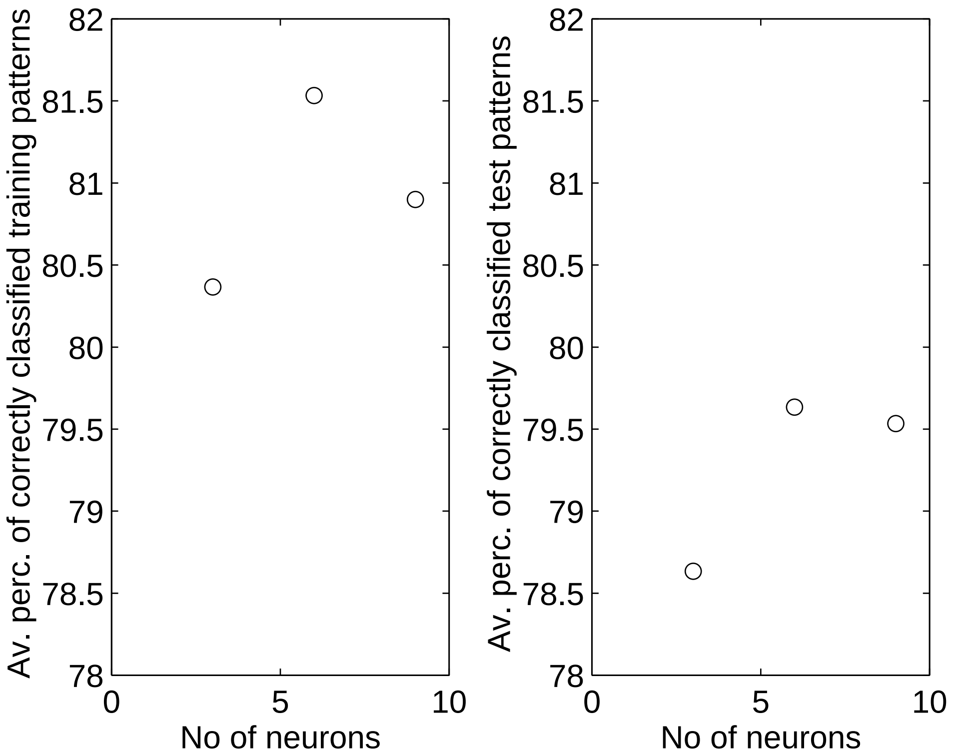

Figure 5.8: Average percentages of correct classifications for training and test using different numbers of output neurons (Perceptron)

The HONN approach using Backpropagation

The 2457 training signals were, in a round robin manner, fed to the backpropagation network. For two training sessions the settings of backpropagation were initialized as follows:

Session number	1	2
No of output neurons	3	3
Learning rate	0.001	0.001
Momentum	0.1	0.05
Steepness	1.0	1.0
No of hidden layers	1	1

The network was trained for different numbers of hidden neurons. Depending on the number of hidden neurons training took 285 to 460 epochs for session one and 428 to 684 epochs for session two. After each training session the 2457 test signals were fed to the network. Figure 5.9 shows the average percentage of correctly classified training and test signals respectively (momentum equals 0.1). Figure 5.10 shows these percentages when the momentum is set to 0.05.

For backpropagation also test sessions were run with constant momentum and varying learning rates, whereby the learning rates were much higher (between 0.01 and 0.99). Results of these test sessions, however, cannot be used as the error of the backpropagation network did not converge. The network still fluctuated heavily after 600 epochs of training.

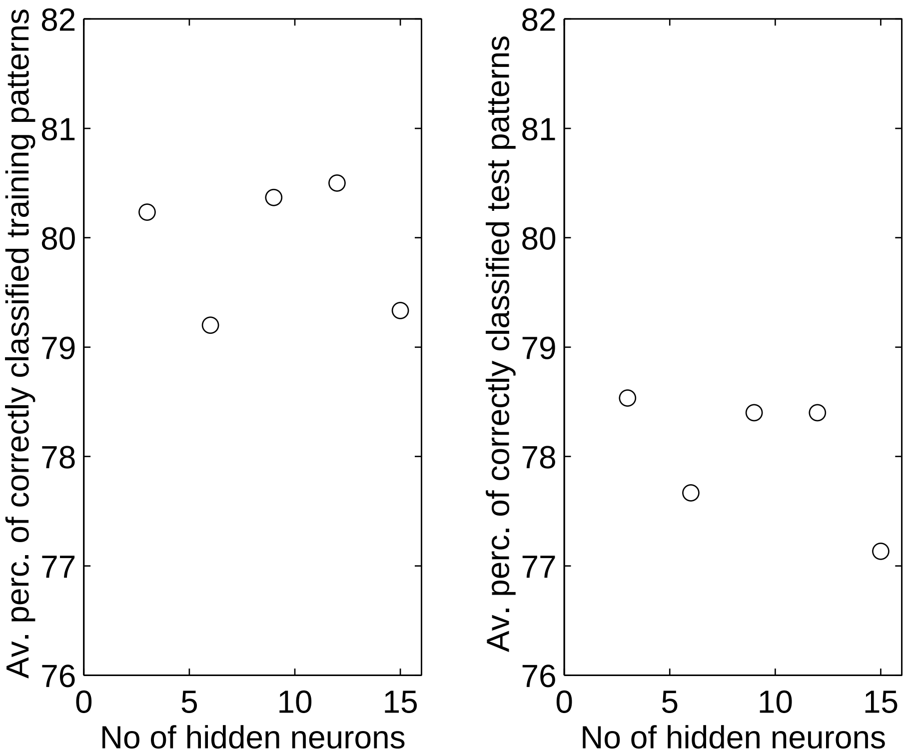

Figure 5.9: Average percentages of correct classifications for training and test using different numbers of hidden neurons (Backpropagation session 1)

5.6.2 Results of session 2

Comparing the signals used for this session with those of session one, it can be seen that only the Gauss filter parameters for class 0 have been changed. With its new parameters the signals of class 0 now were between those of class 1 and class 2.

Class	Sigma	Mean	No of kernels	Multiplication factor
0	5	10	16	0.55
1	5	10	21	0.5
2	5	10	11	0.6

As the signals of this test set are now much closer to each other and even partially overlap, the neural networks should have many more problems (compared with session one) to distinguish between the three classes correctly.

In this session only the HONN approaches using LVQ and backpropagation are used, as both of these neural networks had so much trouble training this test set correctly, that it did not look meaningful also to train the perceptron on this test set.

The HONN approach using LVQ

The 2457 training signals were, in a round robin manner, fed to the LVQ network. For training the settings of LVQ were initialized as follows:

Learning rate	0.01
Learning rate change	0.95

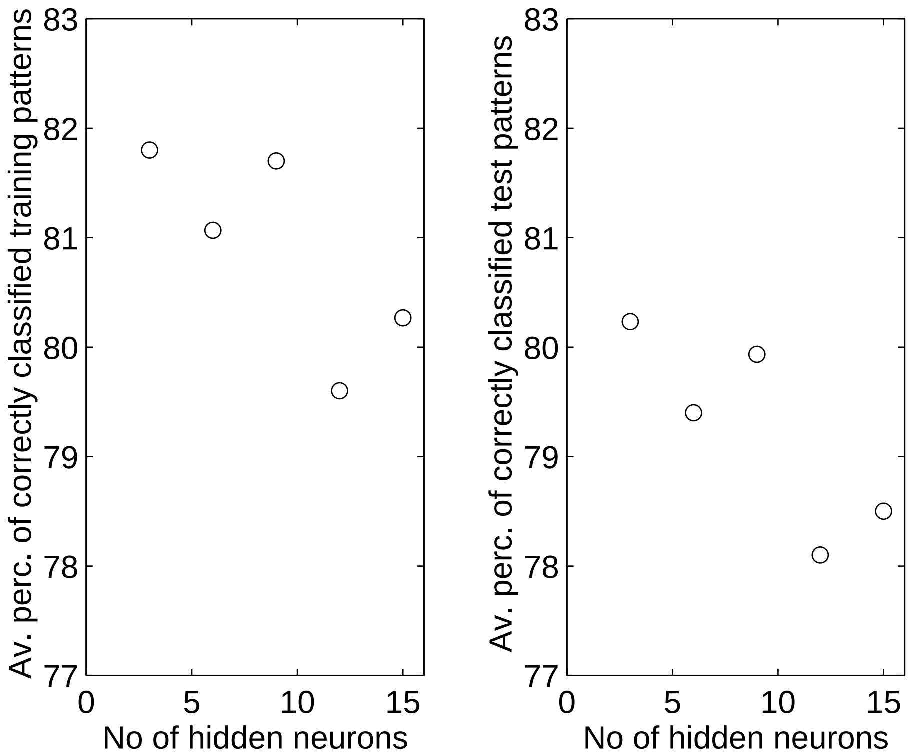

Figure 5.10: Average percentages of correct classifications for training and test using different numbers of hidden neurons (Backpropagation session 2)

With 54 output neurons (the number of output neurons in session one presenting the average optimum for the average percentage of correctly classified patterns in a reasonable learning time) the network was trained in 118 epochs. This showed an average percentage of correctly classified training patterns of 57.4%.

After the training session the 2457 test signals were fed to the network. This yielded an average percentage of correctly classified test patterns of 54.8%.

LVQ had great difficulty training and recognizing the signals of class 2 correctly. Only 37.7% of the training signals and 35.9% of the test signals were classified correctly.

The HONN approach using Backpropagation

The 2457 training signals were, in a round robin manner, fed to the backpropagation network. For this session the settings of backpropagation were initialized as follows:

No of output neurons	3
Learning rate	0.001
Momentum	0.05
Steepness	1.0
No of hidden layers	1

With 9 hidden neurons (the number of output neurons in session one presenting the average optimum for the average percentage of correctly classified patterns in a reasonable learning time) the network was trained in 564 epochs. This led to an average percentage of correctly classified training patterns of 53.0%.

After the training session the 2457 test signals were fed to the network. This resulted in an average percentage of correctly classified test patterns of 51.8%.

Backpropagation had even greater difficulty training and recognizing the signals of class 2 correctly. Only 10.6% of the training signals and 10.0% of the test signals were classified correctly.

5.7 Comparison of the results

In this section the results of the HONN and co-occurrence approaches will be compared. For this comparison the Mahalanobis distance, which will be explained first, will be used as a measure ([Morrison, 1967], [McLachlan, 1992]).

5.7.1 The Mahalanobis distance

Mahalanobis distance

We have obtained many independent random samples of observations on some multidimensional variate. For each pair of classes it is now desired to test the null hypothesis

$$H_0 : \mu_1 = \mu_2$$

that the population mean vectors are identical, as opposed to the alternative hypothesis

$$H_1 : \mu_1 \neq \mu_2$$

of different means. From the data mean vectors have been computed, and the pooled estimate S of the common covariance matrix has been prepared (table below).

Sample size	Sample 1	Sample 2
	N_1	N_2
Mean vector	$\bar{x}_1^T = [\bar{x}_{11}, \cdots, \bar{x}_{1p}]$	$\bar{x}_2^T = [\bar{x}_{21}, \cdots, \bar{x}_{2p}]$
Matrix of sums of squares and products	A_1	A_2
Pooled covariance matrix	$S = \frac{1}{N_1+N_2-2}(A_1 + A_2)$	

Now the Mahalanobis distance equals:

$$D_M = \frac{1}{2}((\bar{x}_1 - \bar{x}_2)^T S^{-1}(\bar{x}_1 - \bar{x}_2))^{\frac{1}{2}}$$

Finally, using [Morrison, 1967] page 307, table 1 'Cumulative Normal Distribution function', the percentages of correct distinction between each two populations can be looked for.

5.7.2 Comparing the results of session 1

Before calculation of the Mahalanobis distance could take place for the results of the HONN approaches using perceptron and LVQ, a small problem had to be solved. Both perceptron and LVQ had a multiple of the number of classes as output neurons. The problem was what to do with the outputs of the output neurons for one class. As most sensible options, the output values could be averaged to one mean or the maximum or the minimum value could be taken.

The decision was made to take the neuron with the minimal output for each class in case of LVQ and the one with the maximum output per class in case of perceptron. This follows the way these two networks handle their classifications, which looked to be the most sensible solution. As backpropagation only had one output neuron per class, for backpropagation no choice had to be made about how to use the output values.

Now, calculating the Mahalanobis distances for the three HONN approaches and the co-occurrence parameters and looking for the percentages of correct distinction, yielded the following results:

LVQ approach

		% correct distinction
$D_M 0-1$	4.7361	100.0%
$D_M 0-2$	5.4474	100.0%
$D_M 1-2$	0.4746	68.2%

Perceptron approach

		% correct distinction
$D_M 0-1$	18.1861	100.0%
$D_M 0-2$	82.4058	100.0%
$D_M 1-2$	0.4702	68.1%

Backpropagation approach

		% correct distinction
$D_M 0-1$	9.1575	100.0%
$D_M 0-2$	9.9047	100.0%
$D_M 1-2$	0.5309	70.2%

Co-occurrence parameters

		% correct distinction
$D_M 0-1$	4.6678	100.0%
$D_M 0-2$	5.4850	100.0%
$D_M 1-2$	0.8351	79.7%

In these tables $D_M x - y$ means the Mahalanobis distance between the classes x and y.

5.7.3 Comparing the results of session 2

To calculate the Mahalanobis distance for the two HONN approaches and the co-occurrence parameters, again the decision was made to take the neuron with the minimal output for each class in case of LVQ. The Mahalanobis distances thus calculated are:

LVQ approach

		% correct distinction
$D_M 0-1$	0.4238	66.3%
$D_M 0-2$	0.1318	55.2%
$D_M 1-2$	0.2942	61.4%

Backpropagation approach

		% correct distinction
$D_M 0-1$	0.7862	78,2%
$D_M 0-2$	0.2137	58.3%
$D_M 1-2$	0.5818	71.9%

Co-occurrence parameters

		% correct distinction
$D_M 0-1$	0.9937	83.9%
$D_M 0-2$	0.1855	57.1%
$D_M 1-2$	0.8351	79.7%

5.8 Conclusions

Looking at the results in the previous section it could be said that generally the co-occurrence parameters seem to do much better than the neural networks. However, as only two tests were done, using signals only chosen to see if the HONN approaches would work, this conclusion would be much too far-reaching at this moment as no thorough test has yet taken place to make clear which of the approaches will be the better.

The advantage of the neural networks compared with the co-occurrence parameters still is the fact that the neural networks are trained on the images while the co-occurrence parameters are just calculated using a predetermined formula.

Therefore, in the next chapter, tests will be done to make clear which approach will be the better.

A conclusion which can be drawn now is that it is very difficult to calibrate the neural networks, especially backpropagation and perceptron. Many of the combinations of parameter settings show a network whose error rate does not converge in a meaningful time or whose percentage of correct classification keeps fluctuating highly during training, when it does converge. As the settings of the network parameters partly depend on the training set, giving correct values to the network parameters will remain a process of trial and error.

Chapter 6

Further tests

6.1 Framework of further tests

In this chapter the HONN approach will be further tested. The theory of section 5.2 will be used to generate test signals. Furthermore, results in these tests will still be compared with results of the co-occurrence parameters using the Mahalanobis distance (section 5.7).

Meaning of the tests in this chapter is to find out under what conditions the neural networks start to get problems training and classifying their data and how the co-occurrence matrix method works under these conditions. Background idea is to trace what differences can still be distinguished by the co-occurrence matrix method and by the neural networks.

Besides, tests are done to look at the influence of the distance in the co-occurrence matrix on the results of the co-occurrence parameters for one set (section 6.3). This has to be done to know what distance is needed to use to make comparison of the HONN results and the co-occurrence parameter results as reliable as possible.

In section 6.4 results of the HONN approaches and the co-occurrence parameters are compared. Therefore, the relative sigma distances are set out against the Mahalanobis distance. The curves thus obtained give an idea of how good the approaches are and at what distances difficulties arise.

Finally, in section 6.5 conclusions drawn from these tests will be provided.

6.2 Further tests on the HONN approach

In this section it is tested how small the sigmas and the difference between the sigmas of the kernels can be made before the neural networks used in the HONN approach start having difficulties training and classifiying their data.

For the HONN approaches the six sets below are used. First, three sets with very small sigmas are used to see how the neural networks react to these sets. Then three test sets with greater sigmas are applied. These sets are generated with only one Gauss kernel and setting the mean equal to zero.

As in the pilot study, for each class 819 training signals and 819 test signals are generated. The number of features of the neural networks used will be 64.

Set 1				
Class	**Sigma**	**Mean**	**No of kernels**	**Multiplication factor**
0	1	0	1	not applicable
1	2	0	1	not applicable
2	3	0	1	not applicable

Set 2				
Class	**Sigma**	**Mean**	**No of kernels**	**Multiplication factor**
0	0.5	0	1	not applicable
1	1.0	0	1	not applicable
2	1.5	0	1	not applicable

Set 3				
Class	**Sigma**	**Mean**	**No of kernels**	**Multiplication factor**
0	0.75	0	1	not applicable
1	1.00	0	1	not applicable
2	1.25	0	1	not applicable

Set 4				
Class	**Sigma**	**Mean**	**No of kernels**	**Multiplication factor**
0	9	0	1	not applicable
1	10	0	1	not applicable
2	11	0	1	not applicable

Set 5				
Class	**Sigma**	**Mean**	**No of kernels**	**Multiplication factor**
0	8	0	1	not applicable
1	10	0	1	not applicable
2	12	0	1	not applicable

Set 6				
Class	**Sigma**	**Mean**	**No of kernels**	**Multiplication factor**
0	6	0	1	not applicable
1	7	0	1	not applicable
2	8	0	1	not applicable

For each of the neural networks the 2457 training signals were, in a round robin manner, fed to them. Initializations for the training of the networks for each test are summarized below.

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
LVQ						
# output neurons	3/6	3/9/54	54/9	54	54	54
Learning rate	0.1	0.1	0.01	0.001	0.01	0.01
L. rate change	0.99	0.99	0.95/0.99	0.95	0.95	0.95
Backpropagation						
# output neurons	3	3	3	-	3	-
# hidden layers	1	1	1	-	1	-
# hidden neurons	3/6	3/6	3/6	-	6	-
Learning rate	0.01	0.01	0.01	-	0.001	-
Momentum	0.1	0.1	0.1	-	0.1	-
Steepness	1.0	1.0	1.0	-	1.0	-
Perceptron						
# output neurons	3/6	3/6	6/9	-	-	-
Learning rate	0.001	0.005	0.001	-	-	-

With each of the neural networks several runs per set were done, each run with another number of neurons (output neurons for LVQ and perceptron and hidden neurons for backpropagation).

With set three, for LVQ the learning rate change had to be changed from 0.95 to 0.99 in the run with six output neurons, because the error rate of the network only increased. In the runs with more output neurons the value for this setting was 0.95.

For set four and set six there are no results for the backpropagation approach. With all the settings and numbers of hidden neurons used, the error rate was still not converging after 3000 epochs and the percentages of correctly classified training patterns still fluctuated highly.

Having had many difficulties with backpropagation fine tuning for set four and perceptron fine tuning for sets one to three, the decision was made not to try the perceptron approach any longer; so for sets four to six there are no results for the perceptron approach.

For the test runs fulfilled, the best results for each set and for each neural network were as follows: for set one the classification results of the recall patterns for all neural networks were between 98.2% and 100.0% per class, while the average for all three classes for all networks was between 99.2% and 100.0%.

With set two these results were between 99.5% and 100.0% with an average percentage between 99.8% and 100.0%.

Also set three gave high percentages; between 93.5% and 100.0% for the individual classes and an average percentage between 96.6% and 98.9%.

For the first three sets the results of the neural networks do not differ much. All of them gave very good results for all sets.

The classification results for LVQ on set four were bad. Especially for class one only 12.9% of the test patterns was classified correctly, which is far below the level of just guessing. The other two classes gave classification results of 78.0% and 78.1% respectively. The average percentage was 56.3%.

The classification results for set five were normal. LVQ presented correct classification percentages between 61.9% and 83.6%, with an average percentage of 72.0% for all three classes in the set. Backpropagation gave percentages between 59.6% and 90.4% with an average percentage of 77.2% for the complete set.

Finally, for set six LVQ gave results between 52.6% and 75.8% with an average percentage of 65.6%.

Not much can be said about the results obtained from these six sets, except for the fact that the neural networks seem to have much more trouble learning the sets with the higher sigma values (four to six).

6.3 The influence of distance on the co-occurrence parameters

For the two test sets below the co-occurrence matrix is calculated with different distances, which is done to study the influence of the distance on the co-occurrence parameters.

Class	Sigma	Mean	No of kernels	Multiplication factor
0	8	0	1	not applicable
1	10	0	1	not applicable
2	12	0	1	not applicable

Class	Sigma	Mean	No of kernels	Multiplication factor
0	6	0	1	not applicable
1	7	0	1	not applicable
2	8	0	1	not applicable

For each distance the co-occurrence parameters are calculated. For these parameters the Mahalanobis distance is calculated. Then the table mentioned earlier is used to find the percentage of correct distinction. The results are summarized in the two figures below.

Looking at the results in figure 6.1 no really optimal distance can be given for use during the calculation of the co-ocurrence matrix. Differences between the results were in the range 0.0% to 0.8%, so these differences are not significant.

Looking at the results of the second test (figure 6.2) the results for distance 5 are best. But the differences between the results of the various distances are in the range of 0.0% to 2.5%; so the choice of the distance to be used for the calculation of the co-occurrence matrix does not present big differences. It can furthermore be observed that for distances smaller than 5 the percentage of correct distinction will increase more. As can be seen the increase will however not be significant

6.4 Comparing the results

After calculating of the co-occurrence parameters for each set the Mahalanobis distances will be computed for all set and all the approaches (co-occurrence and HONN) used. For these distances the percentages of correct distinction will be lloked for. These will be compared with each other in the next subsection.

6.4.1 Comparing results for relative sigma distances

In this subsection the percentages of correct distinction are compared with the relative sigma distances as this will give some nice curves which can tell us how good the approaches really are. This means that for two classes with sigmas 7 and 8 the relative distance $\frac{1}{8}$ is used $((max - min) / max)$. The results of this comparison can be seen in figure 6.3 for LVQ, Backpropagation and co-occurrence matrix method respectively.

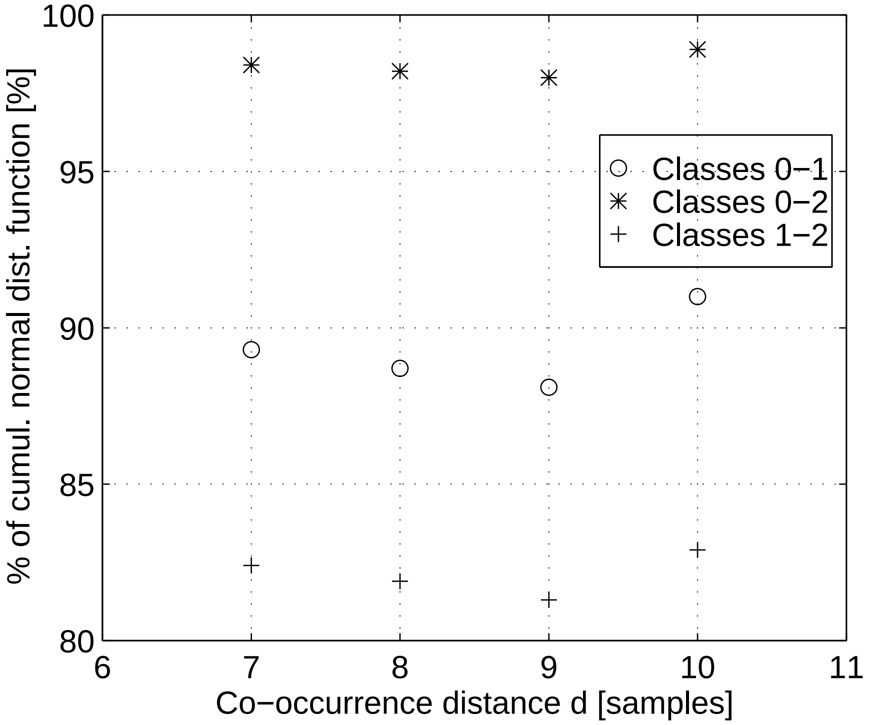

Figure 6.1: Percentages of correct distinction as function of co-occurrence matrix distance for the first set ($Classes\ x - y$ means percentage of cumulative normal distribution function between classes x and y)

For the smaller distances no values are available for backpropagation as this neural network did not converge correctly.

Looking at figure 6.3 it can be seen that for a relative distance larger than $\frac{1}{4}$ LVQ, backpropagation and the co-occurrence matrix method all give a percentage of correct distinction between 85.0% and 100.0%. For backpropagation this already applies above a relative distance of $\frac{1}{6}$.

Looking at the results in this section no conclusion can yet be drawn about which approach is best. Each set has a different approach which distinguishes the set better than the other approaches. Besides, it can not yet be concluded whether, generally, the neural networks give better results than the co-occurrence parameters or not.

Looking at the values between 0 and $\frac{1}{4}$ for LVQ and the co-occurrence matrix method, it can be seen that the performance of the co-occurrence parameter method is slightly better than that of LVQ. LVQ has greater difficulty making a distinction for small relative sigma distances than the co-occurrence parameters.

6.5 Conclusions

Looking at the results obtained not much can be said. The neural network approaches and the co-occurrence parameters are competing with each other on which approach is the better. For some sets the co-occurrence parameters yield better results while for other sets the neural networks perform better.

A conclusion which can be drawn is that the neural networks have many difficul-

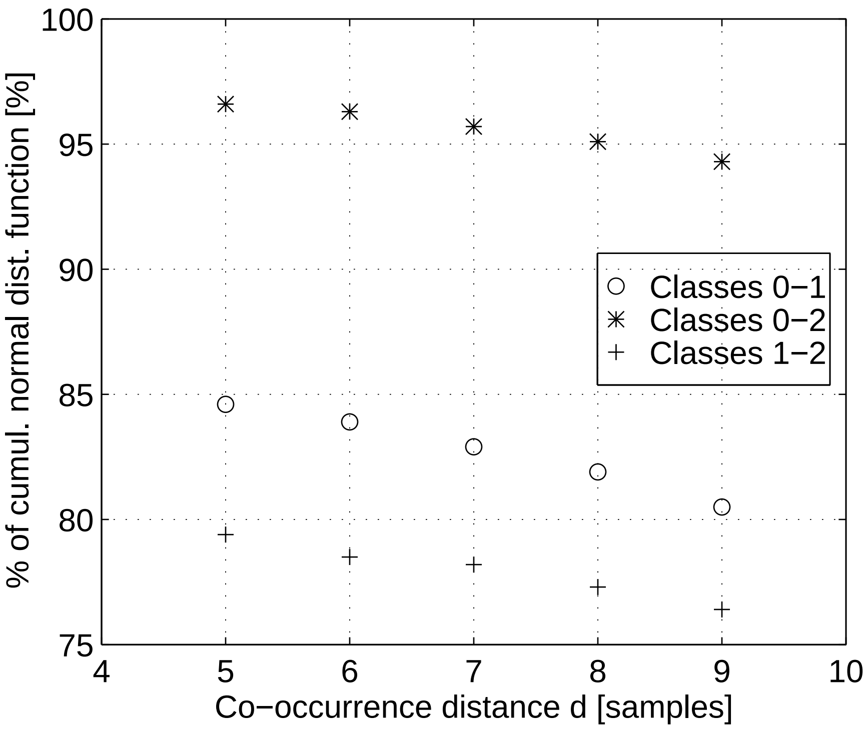

Figure 6.2: Percentages of correct distinction as function of co-occurrence matrix distance for the second set ($Classes\ x - y$ means percentage of cumulative normal distribution funtion between classes x and y)

ties distinguishing classes with small relative sigma distances. Especially backpropagation would not train correctly when the relative sigma distance became smaller than $\frac{1}{6}$.

Looking at figure 6.3 it can be seen that all approaches obtain a 100.0% distinction between the classes when the relative sigma distance is higher than $\frac{1}{3}$.

In the range $\frac{1}{5}$ to $\frac{1}{3}$ backpropagation performed best, still obtaining 100.0% distinction, while the co-occurrence parameters obtain a distinction between 91.0% and 100.0% and LVQ obtains a distinction between 78.0% and 100.0%.

In the range $\frac{1}{10}$ to $\frac{1}{5}$ the co-occurrence parameters still obtain 68.0% to 87.0% correct distinction while LVQ obtains a correct distinction between 61.0% and 86.0%, although most results are between 75.0% and 78.0%.

On an average it has to be said that the co-occurrence parameters deliver slightly better results than the neural networks, but these results are not significantly better. Therefore, it is still not sure whether the neural networks will perform better than the co-occurrence parameters or not. To find this out in the next chapter all approaches will be tested on one-dimensional, simulated images.

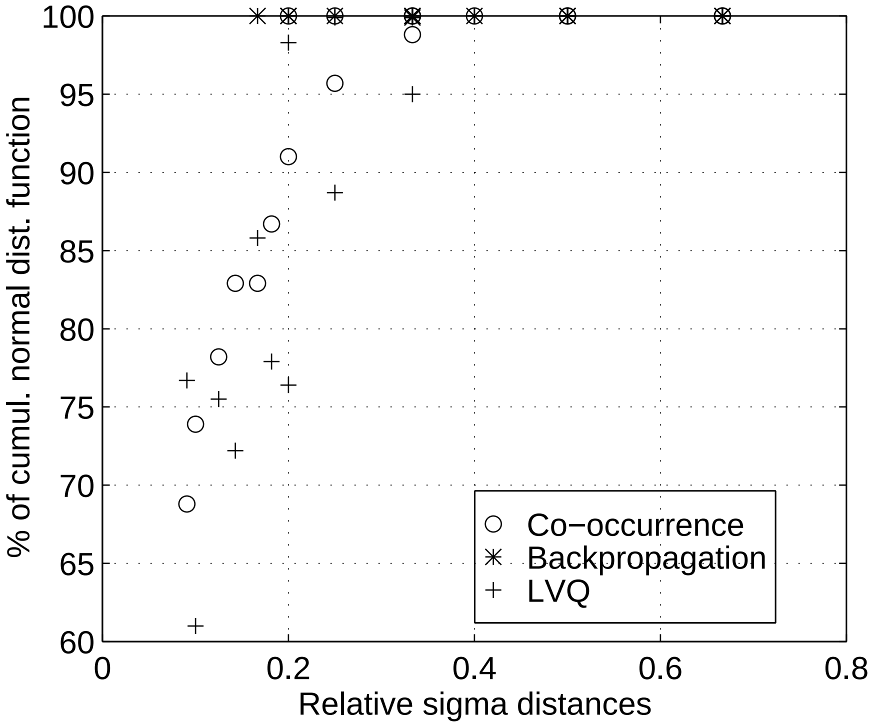

Figure 6.3: The relative sigma distances against the percentage of correct distinction

Chapter 7

Tests on 1-d images

7.1 Introduction to the 1-d image tests

In this chapter tests on one-dimensional, simulated images will be dealt with. Their generation will be explained in section 7.2.

Then, in section 7.3 the results of the neural networks on these images will be given.

In section 7.4 one of the test sets is used to find an optimal distance which can be used to calculate the co-occurrence parameters for all sets.

The results of the HONN approaches will be compared with those of the co-occurrence matrix method in section 7.5.

In section 7.6 the test methods used will be checked for the influence of first-order statistics. No influence of first-order statistics is allowed as especially the co-occurrence parameters will profit from this influence, yielding results which are better than they really should be.

Section 7.7 is an extension to the tests done so far. Backpropagation will be trained on the test sets using two hidden layers instead of one to find out if backpropagation is better prepared for the job this way.

For backpropagation with two hidden layers in section 7.8 the steepness of the activation (being constant during all the tests done) will be varied to see if this can improve the results.

Finally in section 7.9 conclusions on these tests will be drawn.

7.2 Generation of one-dimensional images

Now more realistic tests will be effected, which means that the HONN approaches will be tested on one-dimensional, simulated images. These images come nearer to reality as they are generated using real-world parameters. Furthermore, they contain structure. This was not the case with the one-dimensional signals used in the previous tests. The theory of section 5.2 still holds for the generation of these images.

In this section the parameters used in the generation process are shown.

7.2.1 Parameters used for generation

The parameters used in the generation process can be split into three categories:

Real-world parameters

1. **Real-world parameters**
   The following parameters belong to this category. They are given together with the symbol used for them and their real-world value.

Sample frequency of echo signal received:	fs = 15e6 Hz
Central frequency of the pulse transmitted:	fc = 3.75e6 Hz
Bandwidth of the pulse transmitted:	bw = 2e6 Hz
Speed of sound:	c = 1540 m/s
Number of scatterers in pulse volume:	dens = 10
Power ratio of struct/diffuse scattering:	strength = <will be varied>
Distance ratio of struct/pulse length:	dist = 1.5
Percentage positional noise on struct:	posnoise = 0.20
Quantization in N bits:	N = 8

Simulation parameters

2. **Simulation parameters**
   The following parameters belong to this category. They are given together with the symbol used for them and their value during the generation process.

Sample frequency during simulation:	fsim = 8 * fs
Number of echo samples:	nsamp = 128
Number of rf-lines:	nlines = 8190

Derivative parameters

3. **Derivative parameters**

   sigma = 0.5 / bw
   dt = 1 / fs
   dtsim = 1 / fsim
   nsampsim = nsamp * fsim / fs

7.3 Results of the HONN approaches

In the tests on the one-dimensional images only the LVQ approach and the back-propagation approach are used. Each of the tests is done using 3 classes. The only parameter which is varied is the power ratio of structural/diffuse scattering. The values used for this parameter during the three tests are summarized in the tables below:

Set 1

Class	Power ratio struct/diffuse scattering
0	0.0
1	0.5
2	1.0

Set 2

Class	Power ratio struct/diffuse scattering
0	0.50
1	0.75
2	1.00

Set 3

Class	Power ratio struct/diffuse scattering
0	0.0
1	1.0
2	2.0

Again for each class 819 training signals and 819 test signals are generated. The number of features of the neural networks used is 64.

For each neural network the 2457 training signals were, in a round robin manner, fed to them. In the table below for each set the initial settings for LVQ and backpropagation are given.

	Set 1	Set 2	Set 3
LVQ			
# output neurons	6/12/54	54	54
Learning rate	0.01	0.01	0.01/0.001/0.01
L. rate change	0.95	0.95/0.99	0.95/0.95/0.99
Backpropagation			
# output neurons	3	3	3
# hidden layers	1	1	1
# hidden neurons	54	81	18
Learning rate	0.001	0.0001	0.001
Momentum	0.01	0.1/0.01	0.1
Steepness	1.0	1.0	1.0

Several runs were done for each set varying the number of neurons and the settings of the parameters to find optimal results.

For set one LVQ presented correct classification results for the test patterns which were between 88.8% and 100.0%, while backpropagation gave a correct classification between 82.7% and 100.0%.

With set two both LVQ and backpropagation had great difficulty, especially in correctly classifying class one. LVQ gave a correct classification of 40.4% to 44.6% for this class, while backpropagation gave even worse results: between 10.3% and 21.2%. Classification results of the other two classes were between 72.4% and 85.5% for LVQ and between 70.1% and 86.6% for backpropagation.

LVQ yielded average results for set three: between 60.7% and 100.0%. Backpropagation however, presented worse results: between 48.8% and 100.0%. Again correct classification of class one caused much trouble.

Both neural networks had trouble to correctly distinguish class one from two, especially when the classes were close to each other. On the average LVQ performed slightly better than backpropagation.

Another aspect which attracted attention was the fact that generalization of both LVQ and backpropagation in all three tests was very good. Both LVQ and backpropagation gave, for all three tests, exactly the same outcome for the training phase and the testing phase, meaning that generalization equals 100%.

7.4 The influence of distance on the co-occurrence parameters

As it was unknow what distance should be chosen in the calculation of the co-occurrence matrix, this matrix was calculated with different distances on the test set below.

Class	**Power ratio struct/diffuse scattering**
0	0.0
1	0.5
2	1.0

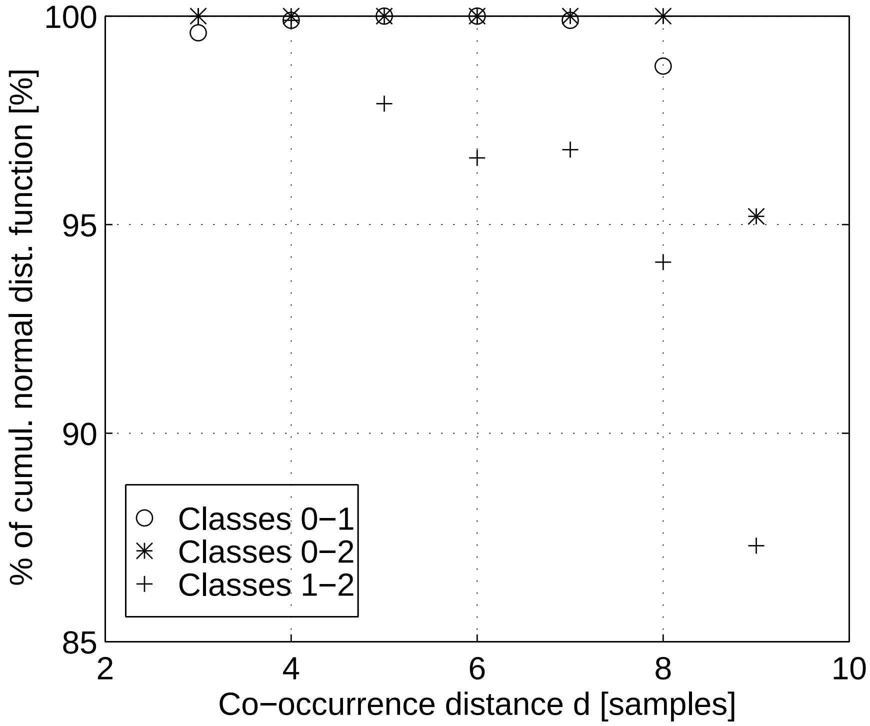

Figure 7.1: Percentages of correct distinction as function of co-occurrence matrix distance (*Classes* $x-y$ means percentage of cumulative normal distribution function between classes x and y). Division of the samples by a factor 8 gives the axial speckle size

The results in figure 7.1 show that at greater distances the percentages of correct distinction start to decrease; so the optimum has to be found in the lower distances (3 to 5). For these lower distances the results of the Mahalanobis distance are almost equal. Therefore, during the comparison of the results in the rest of this chapter the co-occurrence matrix will be calculated with distance 5.

7.5 Comparing the results

The results of the tests done are again compared with each other. For this purpose, the Mahalanobis distance is calculated for all approaches. The Mahalanobis distances and the percentages of correct distinction derived from these Mahalanobis distances are compared below for each set.

Set 1	Classes 0-1		Classes 0-2		Classes 1-2	
LVQ	3.8473	100.0%	5.8934	100.0%	3.3663	99.9%
Backpropagation	8.1104	100.0%	11.2961	100.0%	1.4716	92.9%
Co-occurrence	3.7749	100.0%	5.1282	100.0%	2.0437	97.9%

For the first set the results are more or less equal. LVQ comes out best followed by the co-occurrence parameters and backpropagation. However, the distances are not so big that it can be said that one of the approaches is much better than the others.

Set 2	Classes 0-1		Classes 0-2		Classes 1-2	
LVQ	0.8640	80.5%	1.2236	88.9%	0.3822	64.8%
Backpropagation	0.9638	83.2%	1.3543	91.2%	0.4072	65.9%
Co-occurrence	1.2098	88.7%	2.0437	97.9%	1.0081	84.3%

For set two the results of the co-occurrence parameters are significantly better than those of the neural networks. The results of the neural networks are more or less equal.

Set 3	Classes 0-1		Clases 0-2		Classes 1-2	
LVQ	5.2311	100.0%	6.0808	100.0%	3.0223	99.9%
Backpropagation	12.1878	100.0%	12.8519	100.0%	0.4409	67.0%
Co-occurrence	5.1282	100.0%	5.9204	100.0%	2.7645	99.7%

Also with set three it cannot be said which approach is best. Except for the fact that backpropagation had great difficulty in distinguishing between class zero and class two, the results are equal.

Looking at the results of these tests it again has to be said that the approaches do not differ much. So it is still not clear which approach is best.

7.6 First-order statistics and co-occurrence parameters

In this section it is checked whether the signals contain some first-order statistics. If this is the case, these first-order statistics would have a positive influence on the co-occurrence parameters meaning that they would show better results than they actually have to show.

To check this, the mean and standard deviations from set two are calculated, and the z-transformation of the signals is taken:

$$z = \frac{x - \mu}{\sigma}$$

where

x is the signal.
μ is the mean of x.
σ is the standard deviation of x.

For the signal classes in the original sets there may not be a significant difference between the overall mean values of the classes. Also, there may not be a significant difference between the overall standard deviation values of the classes. The same has to hold for the signal classes after the z-transformation. As can be seen in the table below this restriction holds; so mean and standard deviations have no influence on the co-occurrence parameters.

		Original	**After z-transformation**
class 0	Av. mean	65.7187	98.9976
	Av. st. dev.	35.9697	40.4516
class 1	Av. mean	65.9585	94.9968
	Av. st. dev.	37.6482	43.1103
class 2	Av. mean	66.4168	91.5734
	Av. st. dev.	38.9104	45.2081

To make completely sure that there will be no influence of the first-order statistics also the co-occurrence parameters are calculated for the set after z-transformation. Furthermore, this set is trained by both LVQ and backpropagation. From the results of the co-occurrence parameters, LVQ and backpropagation the Mahalanobis distance is calculated. The Mahalanobis distances may not show significant differences with the Mahalanobis distances of the original set. The test are done with the same parameter settings as used for the original set.

LVQ approach

	Original		**After z-transformation**	
		% correct distinction		% correct distinction
$D_M 0-1$	0.8640	80.5%	0.8698	80.8%
$D_M 0-2$	1.2236	88.9%	1.2675	89.6%
$D_M 1-2$	0.3822	64.8%	0.4081	65.9%

Backpropagation approach

	Original		**After z-transformation**	
		% correct distinction		% correct distinction
$D_M 0-1$	0.9638	83.2%	0.8294	79.4%
$D_M 0-2$	1.3543	91.2%	1.1699	87.9%
$D_M 1-2$	0.4072	65.9%	0.3453	63.7%

Co-occurrence parameters

	Original		**After z-transformation**	
		% correct distinction		% correct distinction
$D_M 0-1$	1.2098	88.7%	1.1406	87.3%
$D_M 0-2$	2.0437	97.9%	1.9047	97.1%
$D_M 1-2$	1.0081	84.3%	0.9105	81.9%

As can be seen from the tables above, the differences of the Mahalanobis distances of the original set and the set after z-transformation are not significant for the three approaches.

Therefore, it can be concluded that the first-order statistics have no influence on the calculation of the co-occurrence parameters.

7.7 Backpropagation with two hidden layers

In this section further tests are reported on backpropagation as the results by this method on the lasts tests were promising.

All tests done with backpropagation so far were carried out with one hidden layer. In this section tests will be done in which backpropagation has two hidden layers. The number of neurons in each hidden layer and the values of the various parameters will be varied to obtain the best results. In the table below settings for the three tests are shown.

	Session 1	**Session 2**	**Session 3**
Backpropagation			
# output neurons	3	3	3
# hidden layers	2	2	2
# neurons layer 1	30	30	64
# neurons layer 2	15	15	32
Learning rate	0.001	0.0001	0.0001
Momentum	0.01	0.1	0.01
Steepness	1	5	5

The three classes of the training and the test sets are generated with the following settings:

Class	**Power ratio struct/diffuse scattering**
0	0.50
1	0.75
2	1.00

As with an increasing number of hidden layers and an increasing number of neurons per hidden layer backpropagation is slowing down rapidly, it is decided only to use 100 signals per class instead of the 819 signals used per class so far. Thus, by reducing the total number of signals to 300 backpropagation is speeded up a bit. The number of features of the neural networks still is 64. The 300 training signals were, in a round robin manner, fed to backpropagation. Each session took between 4100 and 6200 epochs.

Results obtained for session one are not too good. Especially for class one correct classification of the test patterns was 21.0% which is very bad. Class zero and class two had a correct classification percentage of 94.0% and 88.0% respectively.

Results of sessions two and three were normal; correct classification percentages are between 61.0% and 90.0% for all the classes.

After calculation of the Mahalanobis distances for these results it can be seen (figure below) that the results for session three are, on average, best, although not significantly better.

Comparing these results with the results of the co-occurrence parameters for this test set, it can be seen that the distinction of classes zero and one and zero and two is done better by backpropagation. However, in the distinction of classes one and two the co-occurrence parameters yield results which are about 10% better, which is significantly better.

A careful conclusion at this moment is that it looks as if the results of backpropagation are becoming better with two hidden layers, an increasing number of neurons in the hidden layers and an increasing value of the steepness of the activation function. As increasing the number of neurons in the hidden layers only makes the network much slower and does not make much sense when we look at the number of input neurons (64), in the next section the influence of the steepness of the activation function will be looked at.

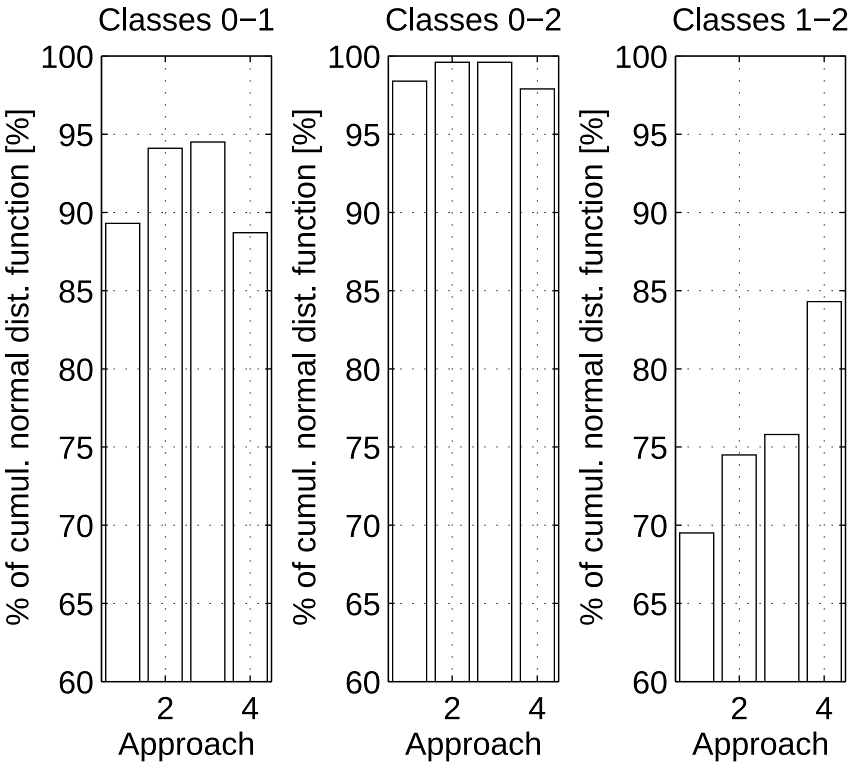

Figure 7.2: Backpropagation results compared with co-occurrence results. In all graphs bar 1 to 3 are results of three backpropagation tests; bar 4 presents the results of the co-occurrence parameters on the same set

7.8 Changing the steepness of the activation

For all tests done so far the steepness of the activation function of backpropagation is kept constant t 1.0. In the test sessions of the previous section, however, it can be seen that this steepness may have some positive influence on the results, meaning that there is a possibility that the results of backpropagation will become better than those of the co-occurrence parameters. Therefore, in the tests done in this section only the steepness of the activation function will be varied. The set used for this test is:

Set 2	
Class	**Power ratio struct/diffuse scattering**
0	0.50
1	0.75
2	1.00

The parameter settings for backpropagation are shown below. As can be seen only the steepness of the activation (psi) is varied.

	Session 1	Session 2	Session 3
Backpropagation			
# output neurons	3	3	3
# hidden layers	2	2	2
# neurons layer 1	64	64	64
# neurons layer 2	32	32	32
Learning rate	0.0001	0.0001	0.0001
Momentum	0.01	0.01	0.01
Steepness	5	6	7

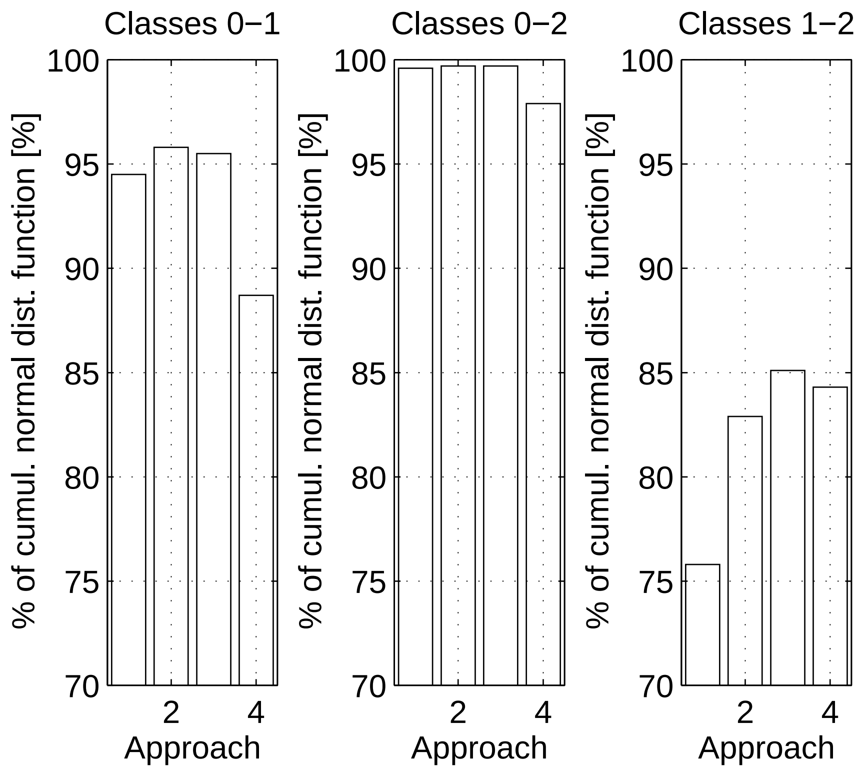

Figure 7.3: Backpropagation results compared with co-occurrence results. For backpropagation only the steepness of activation is changed. In all graphs bar 1 to 3 are results of three backpropagation tests; bar 4 presents the results of the co-occurrence parameters on the same set

From the results in figure 7.3 it can be seen that the results for a steepness of activation equal to 7 are the best. Furthermore it can be seen that backpropagation for this value gives better results in the distinction of all couples of classes. Especially the distinction between classes 0 and 1 is much better.

To find out if the results of backpropagation with this steepness setting are always better than the results of the co-occurrence parameters, tests have been done with some more sets shown below.

Set 1

Class	Power ratio struct/diffuse scattering
0	0.0
1	0.5
2	1.0

Set 2

Class	Power ratio struct/diffuse scattering
0	0.0
1	1.0
2	2.0

Set 3

Class	Power ratio struct/diffuse scattering
0	0.2
1	0.6
2	1.0

Set 4

Class	Power ratio struct/diffuse scattering
0	0.8
1	0.9
2	1.0

Set 5

Class	Power ratio struct/diffuse scattering
0	0.5
1	0.6

Set 6

Class	Power ratio struct/diffuse scattering
0	0.75
1	1.00

Set 7

Class	Power ratio struct/diffuse scattering
0	0.8
1	1.0

Set 8

Class	Power ratio struct/diffuse scattering
0	0.7
1	1.0

From the backpropagation results of these sets only the results of those sets are used which achieved a generalization of 100.0%. This caused some of the results to be dropped as backpropagation did not achieve 100.0% generalization.

The tests on these sets are done with different numbers of classes in the training and test sets. Also different numbers of signals per class are used (ranging from 50 signals per class up to 300 signals per class).

Then from the co-occurrence and the backpropagation results the Mahalanobis distance is calculated, which is then set out against the relative strength between the classes compared. For strengths 0.8 and 1.0 the relative strength equals $\frac{1}{5}$. This is calculated with the formula $((max - min) / max)$.

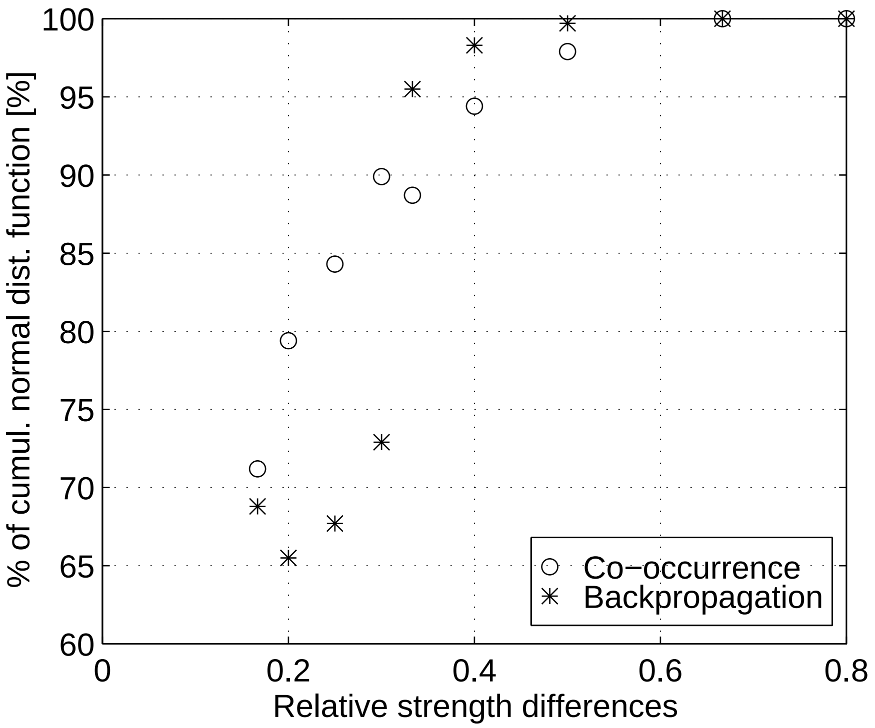

Figure 7.4: The relative strength against the percentage of correct distinction

As can be seen from figure 7.4, in the range $\frac{3}{10}$ to $\frac{1}{2}$ backpropagation performs about 2% to 7% better than the co-occurrence parameters.

For relative strength differences higher than $\frac{1}{2}$ both approaches render equally good results.

For relative strength differences smaller than $\frac{3}{10}$ the performance of backpropagation with two hidden layers decreases rapidly, while the performance of the co-occurrence parameters only decreases slowly, making the co-occurrence parameters performing much better in this range.

A similar conclusion can be drawn from figure 7.5 in which the percentages of correct distinction of backpropagation are set out against those of the co-occurrence parameters. For higher percentages of correct distinction, backpropagation performs better while for percentages below 85% the co-occurrence parameters perform better.

7.9 Conclusions

Looking at the results in this chapter it must again be said that, generally, the co-occurrence parameters yield slightly better results than LVQ and backpropagation.

Only when backpropagation with two hidden layers is used and the steepness of the activation is increased, backpropagation is about 2% to 7% better than the co-occurrence parameters over a small range of relative strength differences.

When the relative strength differences decrease, also the perormance of back-propagation with two hidden layers decreases rapidly, while the performance of the

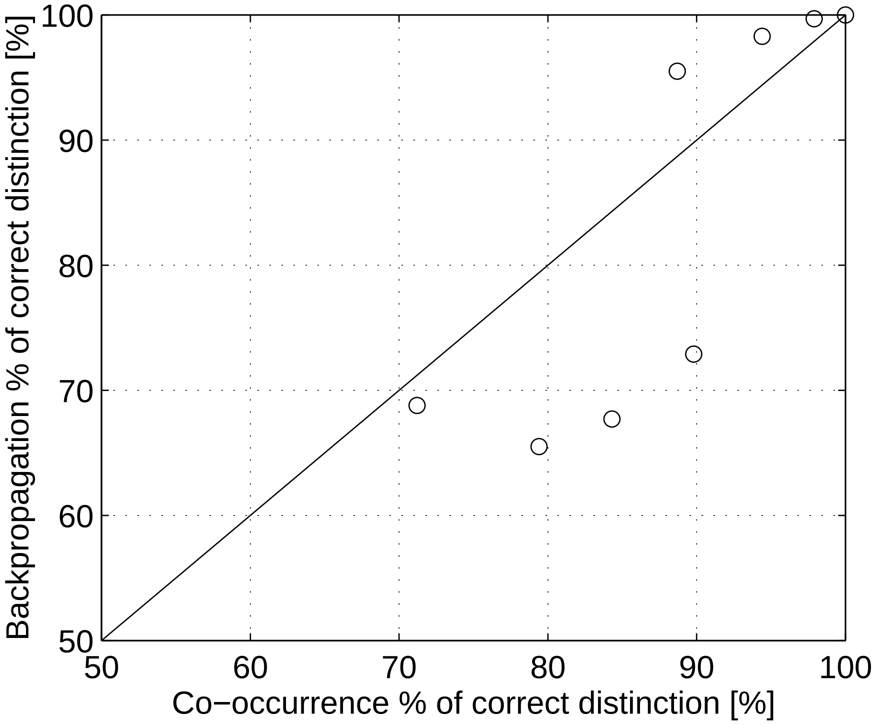

Figure 7.5: Backpropagation correct distinction versus co-occurrence parameters correct distinction

co-occurrence parameters only decreases slowly, making the co-occurrence parameters better for very small relative strength differences.

Again there were difficulties in the calibration of the neural networks; so many tries had to be made before a combination of parameter settings was found which made the neural network train correct.

One more thing can be mentioned: when the neural networks were trained on the one-dimensional images, generalization of the test sets was in most cases equal to 100.0%. This is really very good.

Chapter 8

Conclusions and recommendations

8.1 Introduction

Now that all tests have been done and results have been worked out, this chapter will deal with what approach can be used best.

Conclusions are divided into three parts:

1. Conclusions concerning the neural networks
2. Conclusions concerning neural networks on simulated rf-images
3. Conclusions concerning the best approach

Finally, some recommendations for future research are made.

8.2 Conclusions

In this section conclusions will be given about the neural networks used and the use of these neural networks to calculate parameters from simulated rf-images.

Furthermore, a discussion is made on which approach to use in certain cases.

8.2.1 Conclusions concerning the neural networks

During all the tests done with the neural networks two facts came up quite painfully:

1. The neural networks used in this thesis are very hard to specify. For one test set to be trained correctly many tests had to be run before correct parameter settings were found so that the error rate converged and the correct classifications on the diagonal of the confusion matrix were not fluctuating heavily.
2. It was hard to train the networks once correct parameter settings were found. Training could go on for hundreds of epochs with a nicely decreasing error rate when suddenly the error rate started to fluctuate heavily, making the results unuseful.

These problems are known to many researchers and can be found in literature. For some simple problems there will be solutions at hand, but for other research areas, like the one in this thesis, these two problems have to be solved in a process of trial and error and with the experience and the feeling for the problem of the researcher.

8.2.2 Conclusions concerning neural networks applied to simulated rf-images

The Higher-Order Neural Networks in this thesis were used in a different way from the one presented in literature: the autocovariance was not built into the weight vectors of the neural network but fed to the inputs of the networks. For the neural network part, beside perceptron, also LVQ and backpropagation were used.

In principle all three approaches of the HONN (perceptron, LVQ and backpropagation) can be used as self-learning image parameters (chapter 5). When more tests are done the perceptron approach is the first one which starts to get many difficulties. This could have been expected before, as perceptron can only be used for linearly separable problems. As the autocovariance function obtained from the rf-images is not linear (second-order) the perceptron approach is the first approach which shows not to be capable as self-learning image parameter.

The LVQ and backpropagation (with one hidden layer) approaches showed to have the power to create self-learning parameters from the images. This fact followed the expectation that these two neural networks would be good as self-learning image parameters. As can be found in literature both LVQ and backpropagation should be capable of learning non-linear problems.

Comparing the results of LVQ and backpropagation with one hidden layer with the results of known statistical parameters (in this thesis four co-occurrence parameters), none of these two HONN approaches and the co-occurrence parameters is really better than the other two. For many test sets the co-occurrence parameters are slightly, but not significantly, better than the HONN approaches.

A final HONN approach used was the one using backpropagation with two hidden layers. This approach presented some more useful results. Comparing these results with those of the co-occurrence parameters the following conclusions could be drawn:

1. For big relative differences between the classes both present equally good results.
2. For average relative differences between the classes backpropagation performs about 2% to 7% better than the co-occurence parameters.
3. For small relative differences between the classes performance of backpropagation rapidly decreased, while the performance of the co-occurrence parameters only decreased slowly, making the co-occurrence parameters perform better than backpropagation.

8.2.3 Conclusions concerning the best approach

Looking at the conclusions in the previous subsections it has to be said that, generally, the HONN approaches with LVQ and backpropagation perform equally well as the co-occurrence parameters; so one final conclusion is that these Higher-Order Neural Networks can be used for this kind of research. They can be used as self-learning image parameters.

The second conclusion which can be drawn is that using the backpropagation approach with two hidden layers performs better than the co-occurrence parameters for a small range of classes; so using this approach will improve diagnostics.

8.3 Recommendations for future research

During the tests done many problems arose which need further research:

- Calibration of the neural networks. For almost each test set used new parameter settings were needed. Finding these new parameter settings was a time consuming problem, as many tests had to be done before parameter settings were found which gave a converging error rate and stable classification results.

  More research has to be done to devrease this problem. It would be useful if it would be possible for neural networks to check a group of parameter settings automatically in a parallel way, skipping those sets which do not fulfil a certain condition and going on with those settings which fulfil this condition, finally presenting some sets of parameter settings which can be used for training the neural network on the problem.

- Training takes long. Training time could be reduced by improving the implementation of the neural networks (better written code, code for parallel execution) and using faster (and parallel) hardware. At least on the implementation of the neural networks some research can be done.

- A problem is (and will probably ever be) the fact that large data sets are needed. These are only available with simulated data, not with real data.

  Tests done with too small sets confirmed the fact mentioned by [Foley, 1972] that the number of samples in the data set must at least be three times the number of features in the neural network. With real data it will be very difficult to come up to this criterion and still have a useful neural network which is capable of mastering the problem.

- More research can be done on finding more exact ranges of relative class differences for which backpropagation with two hidden layers will be significantly better than the co-occurrence parameters.

- Research also has to be done to improve the results of backpropagation with two hidden layers both in the area in which it is better than the co-occurrence parameters and in the area in which results are worse than those of the co-occurrence parameters, as it is not known whether for backpropagation optimal parameter settings have been used or not.

Books of Reference

[Bendat et al., 1971] Bendat J.S., Piersol A.G.
Random data: Analysis and measurements procedures
(United States of America, 1971)

[Enden et al., 1987] Enden A.W.M. van den, Verhoeckx N.A.M
Digitale signaalbewerking
(Amerongen, 1987)

[Gonzalez et al., 1992] Gonzalez R.C., Woods R.E.
Digital image processing
(United States of America, 1992)

[Hertz et., 1991] Hertz J., Krogh A., Palmer R.G.
Introduction to the theory of neural computation
(Redwood City, 1991)

[Kohonen, 1984] Kohonen T.
Self-organization and associative memory
(Berlin, 1984)

[Math Works, 1984] The Math Works Inc.
Matlab user's guide for unix workstations
(Natrick, 1984)

[McLachlan, 1992] McLachlan G.J.
Discriminant analysis and statistical pattern recognition
(New York, 1992)

[Morrison, 1967] Morrison D.F.
Multivariate statistical methods
(New York, 1967)

[Oppenheim et al., 1975] Oppenheim A.V., Schafer R.W.
Digital signal processing
(Bell Telephone Laboratories, Inc, 1975)

[Papoulis, 1965] Papoulis A.
Probability, random variables, and stochastic processes
(Singapore, 1965)

[Thijsen et al., 1991] Ackerstaff R.G.A., Asten W.N.J.C. van, Baerts W., Cuijpers M.H.M., Dalen A. van, Daniëls O., Dijk, R. van, Gussenhoven W.J. van, Hoeks A.P.G., Hooge P. de, Lubbers J., Sanders F.B.M., Stoutenbeek Ph., Thijssen J.M., Verbeek A.M., Zonderland H.M.
Handboek echografie
(Leiderdorp, 1991)

Bibliography

[Burckhardt, 1978] Burckhardt C.B.
Speckle in Ultrasound B-Mode Scans
IEEE Transactions on Sonics and Ultrasonics 25(1)
Pages 1-6
1978

[Cho et al., 1992] Cho S.B., Kim J.H.
A two-stage classification scheme with backpropagation neural network classifiers
Pattern Recognition Letters 13
Pages 309-313
1992

[Conners et al., 1980] Conners R.W., Harlow C.A.
A theoretical comparison of texture algorithms
IEEE Transactions on Pattern Analysis and Machine Intelligence 2(3)
Pages 204-222
1980

[Fabel, 1988] Fabel R.
Laterale diffraktiekorrektie van ultrageluidsscans
Afstudeerverslag Biofysisch Laboratorium, Oogheelkunde
1988

[Ferdeghini et al., 1991] Ferdeghini E.M., Pinamonti B., Picano E., Lattanzi F., Bussani R., Slavich G., Benassi A., Camerini F., Landini L., L'Abbate A.
Quantitative texture analysis in echocardiography: application to the diagnosis of myocarditis
Journal of Clinical Ultrasound 19
Pages 263-270
1991

[Foley, 1972] Foley D.H.
Considerations of sample and feature size
IEEE Transaction on Information Theory 18(5)
Pages 618-626
1972

[Gebbinck et al., 1993] Gebbinck M.S. klein, Verhoeven J.T.M., Thijssen J.M., Schouten T.E.
Application of neural networks for the classification of diffuse liver disease by quantitative echography
Ultrasonic Imaging 15
Pages 205-217
1993

[Gerbrands, 1989] Gerbrands, J.J.
Inleiding in de digitale beeldverwerking
Dictaat
(Delft, 1989)

[Giles et al., 1987] Giles C.L., Maxwell T.
Learning, invariance, and generalization in high-order neural networks
Applied Optics 26(23)
Pages 4972-4978
1987

[Giles et al., 1988] Giles C.L., Griffin R.D., Maxwell T.
Encoding geometric invariances in higher-order neural networks
Neural Information Processing Systems
Pages 301-309
1988

[Haberkorn et al., 1990] Haberkorn U., Zuna I., Lorenz A., Zerban H., Layer G., Kaick G. van, Räth U.
Echographic tissue characterization in diffuse parenchymal liver disease: correlation of image structure with histology
Ultrasonic Imaging 12
Pages 155-170
1990

[Haralick et al., 1973] Haralick R.M., Shanmugam K., Dinstein I.
Texture features for image classification
IEEE Transactions on Systems, Man, and Cybernetics 3(6)
Pages 610-621
1973

[Haralick, 1979] Haralick R.M.
Statistical and structural approaches to texture
Proceedings of the IEEE 67(5)
Pages 786-804
(1979)

[Harris et al.] Harris R.A., Wells P.N.T.
Ultimate Limits in Ultrasound Imaging Resolution
Clin. Diagn. Ultrasound 28
Pages 109-123
1993

[Jacobs et al., 1991] Jacobs E.M.G.P., Thijssen J.M.
A simulation study of echographic imaging of diffuse and structurally scattering media
Ultrasonic Imaging 13
Pages 316-333
1991

[Jain et al., 1991] Jain, A.L., Mao J.
A k-nearest neighbor artificial neural network classifiers
Proceedings of the International Joint Conference on Neural Networks
Pages 515-520
(Seattle, 1991)

[Kanaoka et al., 1992] Kanaoka T., Chellappa R., Yoshitaka M., Tomita S.
A higher-order neural network for distortion invariant pattern recognition
Pattern Recognition Letters 13
Pages 837-841
1992

[Layer et al., 1990] Layer G., Zuna I., Lorenz A., Zerban H., Haberkorn U., Bannash P., Kaick G. van, Räth U.
Computerized ultrasound B-scan texture analysis of experimental fatty liver disease: Influence of total lipid content and fat deposit distribution
Ultrasonic Imaging 12
Pages 171-188
1990

[Layer et al., 1991] Layer G., Zuna I., Lorenz A., Zerban H., Haberkorn U., Bannasch P, Kaick G. van, Räth U.
Computerized ultrasound B-scan texture analysis of experimental diffuse parenchymal liver disease: correlation with histopathology and tissue composition
Journal of Clinical Ultrasound 19
Pages 193-201
1991

[Linhart et al., 1992] Linhart G., Dorffner G.
A self-learning visual pattern explorer and recognizer using a higher order neural network
International Joint Conference on Neural Networks
1992

[Lippmann, 1989] Lippmann R.P.
Pattern classification using neural networks
IEEE Communications Magazine, November
Pages 47-63
1989

[Maxwell et al., 1986] Maxwell T., Giles C.L., Lee Y.C., Chen H.H.
Nonlinear dynamics of artificial neural systems
Neural networks for computing
(New York, 1986)

[Nicholas et al., 1986] Nicholas D., Nassiri D.K., Garbutt P., Hill C.R.
Tissue characterization from ultrasound B-scan data
Ultrasound in Medicine and Biology 12(2)
Pages 135-143
1986

[Oosterveld, 1984] Oosterveld B.J.
Textuur in ultrageluids b-mode scans
analyse en bewerking
1984

[Oosterveld, 1990] Oosterveld B.J.
On the quantitative analysis of ultrasound signals with applications to diffuse liver disease
Thesis
Nijmegen University
1990

[Pan et al., 1992] Pan H.L., Chen C.Y.
Liver tissues classification by artificial neural networks
Pattern Recognition Letters 13
Pages 355-368
1992

[Pinamonti et al., 1989] Pinamonti B., Picano E., Ferdeghini E.M., Lattanzi F., Slavich G., Landini L., Camerini F., Benassi A., Distante A., L'Abbate A.
Quantitative texture analysis in two-dimensional echocardiography: application to the diagnoses of myocardial amyloidosis
Journal of the American College of Cardiology 14(3)
Pages 666-671
1989

[Räth et al., 1985] Räth U., Schlaps D., Limberg B., Zuna I., Lorenz A., Kaick G. van, Lorenz W.J., Kommerell B.
Diagnostic accuracy of computerized B-scan texture analysis and conventional ultrasonography in diffuse parenchymal and malignant liver disease
Journal of Clinical Ultrasound 13
Pages 87-99
1985

[Reid et al., 1989a] Reid M.B., Spirkovska L., Ochoa E.
Simultaneous position, scale, and rotation invariant pattern classification using third-order neural networks
Neural Networks 1(3)
Pages 154-159
1989

[Reid et al., 1989b] Reid M.B., Spirkovska L., Ochoa E.
Rapid training of higher-order neural networks for invariant pattern recognition
International Joint Conference on Neural Networks
Pages 1689-1692
1989

[Rutten et al., 1992] Rutten L., Schouten T., Vuurpijl L.
Netwerk modellen en beschrijvingen ervan in Miranda
Dictaat TTN, deel 2
(Nijmegen, 1992)

[Schlaps et al., 1987] Schlaps D., Zuna I., Walz M., Volk J., Räth U., Lorenz A., Kaick G. van, Lorenz W.J.
Ultrasonic tissue characterization by texture analysis: elimination of tissue-independent factors
SPIE 786
Pages 128-134
1987

[Schmidt et al., 1993] Schmidt W.A.C., Davis J.P.
Pattern recognition properties of various feature spaces for higher order neural networks
IEEE Transactions on Pattern Analysis and Machine Intelligence 15(8)
Pages 795-801
1993

[Shawker et al., 1985] Shawker T.H., Garra B.S., Insana M.F.
Ultrasonic Tissue Characterization: Fundamental Concepts and Clinical Applications
Ultrasound Annual 1985
1985

[Silverman et al., 1990] Silverman R.H., Noetzel A.S.
Image processing and pattern recognition in ultrasonograms by backpropagation
Neural networks 3
Pages 593-603
1990

[Spirkovska et al., 1989] Spirkovska L., Reid M.B.
Connectivity strategies for higher-order neural networks applied to pattern recognition
Proceedings of the International Joint Conference on Neural Networks
Pages 121-126
(Piscataway, 1989)

[Thijssen et al., 1990] Thijssen J.M., Oosterveld B.J.
Texture in tissue echograms
Speckle or Information
Journal of Ultrasound in Medicine 9
Pages 215-229
1990

[Thijssen, 1992] Thijssen J.M.
Echographic Image Processing
Advances in Electronics and Electron Physics 84
Pages 317-349
1992

[Vuurpijl, 1994] Vuurpijl L., Schouten T., Vytopil J.
Convis: action oriented control and visualization of neural networks
Introduction and technical description
(University of Nijmegen, 1994)

[Wagner et al., 1987] Wagner, R.F., Insana M.F., Brown D.G.
Statistical properties of radio-frequency and envelope-detected signals with applications to medical ultrasound
Journal of the Optical Society of America 4
Pages 910-922
1987

[Wells] Wells P.N.T.
Advances in imaging techniques
Clin. Diagn. Ultrasound 28
Pages 47-53
1993

[Windsor et al., 1993] Windsor C.G., Anselme F., Capineri L., Mason J.P.
The classification of weld defects from ultrasonic images: a neural network approach
The British Journal of Non-Destructive Testing 35(1)
Pages 15-22
1993