

**Zellulare Nichtlineare Netzwerke –  
Optimierungsverfahren und  
Anwendungen**

**Dissertation  
zur Erlangung des Doktorgrades  
der Naturwissenschaften**

**vorgelegt beim Fachbereich Physik  
der Johann Wolfgang Goethe-Universität  
in Frankfurt am Main**

**von  
Gunter C. Geis  
aus  
Frankfurt am Main**

**Frankfurt am Main, 2012**

vom Fachbereich Physik der Johann Wolfgang Goethe-Universität  
als Dissertation angenommen.

Dekan: Prof. Dr. Huth

Gutachter: Prof. Dr. R. Tetzlaff  
Prof. Dr. A. Lacroix  
Prof. Dr. J. Maruhn

Datum der Disputation: 2. August 2012

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Zellulare Systeme . . . . .	5
2.2	CNN . . . . .	6
2.2.1	Zellenanordnung . . . . .	8
2.2.2	Nachbarschaft . . . . .	8
2.2.3	Randbedingungen . . . . .	9
2.2.4	Templates . . . . .	13
2.2.5	Ausgangsfunktion . . . . .	16
2.2.6	Zustandsgleichung für verschiedene Klassen von CNN . . . . .	16
2.3	Numerische Simulation von CNN . . . . .	22
<b>3</b>	<b>EyeRIS-System 1.1 mit analogem Ace16kv2 Focal- Plane Prozessor</b>	<b>25</b>
3.1	ACE16kv2 und das Full-Range Modell . . . . .	26
3.1.1	Bildverarbeitung mit dem ACE16kv2 . . . . .	31
3.1.2	Ausführungsgeschwindigkeit einzelner Routinen	33
3.2	Darstellungsgenauigkeit von Grauwerten . . . . .	36
<b>4</b>	<b>Musterdetektion mit dem ACE16kv2</b>	<b>39</b>
4.1	Implementierung auf dem EyeRIS System . . . . .	45
4.2	Ergebnisse . . . . .	46
<b>5</b>	<b>Parameteroptimierung</b>	<b>51</b>
5.1	Fehlermaß . . . . .	53
5.2	Optimierungsverfahren für CNN . . . . .	54

5.3	Der SSAPE-Algorithmus auf dem EyeRIS-System . . .	60
5.4	SSAPE mit Graustufenbildern . . . . .	64
<b>6</b>	<b>Statistische Untersuchungsmethoden</b>	<b>69</b>
6.1	Die Receiver Operating Characteristic (ROC) . . . . .	69
6.2	Nullhypotesentest und Surrogate . . . . .	74
<b>7</b>	<b>Anwendungsbereiche</b>	<b>79</b>
7.1	Vorhersage epileptischer Anfälle . . . . .	79
7.1.1	EEG-Signale und epileptische Anfallsvorhersage	85
7.1.2	Datenbasis . . . . .	86
7.1.3	Prädiktion mit DT-CNN . . . . .	89
7.1.4	Ergebnisse der Prädiktion durch Anwendung von Optimierungsverfahren . . . . .	96
7.1.5	Prädiktion durch Mustererkennung und Sym- bolisierung . . . . .	101
7.1.6	Statistik der Musterhäufigkeiten . . . . .	108
7.1.7	Ergebnisse der Musterdetektion . . . . .	110
7.2	Laserschweißen . . . . .	115
7.2.1	Datenbasis . . . . .	118
7.2.2	Algorithmus . . . . .	120
7.2.3	Laufzeitverhalten . . . . .	123
7.2.4	Ergebnisse . . . . .	135
<b>8</b>	<b>Zusammenfassung</b>	<b>141</b>
<b>A</b>	<b>Anhang mit weiteren Untersuchungsergebnissen</b>	<b>145</b>
A.1	Optimierungsverfahren . . . . .	145
A.2	Epilepsievorhersage mit Prädiktor . . . . .	149
A.3	Epilepsievorhersage mit Musterdetektion . . . . .	166
A.4	Key-Hole-Detektion mit digitalen Rechnern und Gra- fikprozessor . . . . .	172
	<b>Danksagung</b>	<b>175</b>
	<b>Literaturverzeichnis</b>	<b>177</b>

<b>Verzeichnis der Tabellen</b>	<b>189</b>
---------------------------------	------------

<b>Verzeichnis der Bildunterschriften</b>	<b>191</b>
---	------------



# 1 Einleitung

Zellulare Nichtlineare Netzwerke bzw. Zellulare Neuronale Netzwerke, sogenannte CNN <sup>1</sup>, wurden 1988 von L.O. Chua und L.Yang [1] eingeführt und seither intensiv untersucht. Durch ihren Aufbau ermöglichen sie eine parallele Daten- und Signalverarbeitung. Die Entwicklung von schaltungstechnischen Realisierungen von CNN – CNN-Hardware – bietet es nun erstmals an, CNN in tragbare Geräte zu integrieren und gewinnen dadurch immer mehr an Bedeutung. Da elektronische Schaltungen durch Rauschen oder andere Einflüsse verursachte Abweichungen von mit Simulationen erlangten Ergebnissen aufweisen, ist es wichtig zu untersuchen, wie die bereits mittels Simulationen entwickelten Algorithmen sich auf CNN-Hardware übertragen lassen.

In dieser Arbeit sollen Algorithmen auf einer schaltungstechnischen Realisierung von Zellularen Nichtlinearen Netzwerken untersucht werden. Hierfür steht das EyeRIS 1.1 System des Unternehmens ANA-FOCUS Ltd. zur Verfügung, welches einen sogenannten Focal Plane Prozessor (FPP), der mit analogen Signalen und Zuständen arbeitet, mit einem digitalen Prozessor kombiniert. Dies ermöglicht es, analoge CNN-Algorithmen zusammen mit digitalen Algorithmen auf einem System zu implementieren und so die Vorteile beider Ansätze zu nutzen. Daten werden hierbei mittels digital/analog (DA)- und analog/-digital (AD)- Wandlung mit einer gleichförmigen Auflösung von 8 bit gewandelt, um den Datenaustausch zu gewährleisten.

---

<sup>1</sup>Englisch: Cellular Neural Network kurz CNN, auch als Cellular Nonlinear Network bezeichnet.

In der vorliegenden Arbeit soll besonderes Augenmerk auf medizinische und technische Anwendungsgebiete gelegt werden.

Im medizinischen Anwendungsbereich soll die Implementierung von Algorithmen zur Vorhersage epileptischer Anfälle untersucht werden. Ziel ist es zu zeigen, daß prinzipiell die Implementierung von CNN-basierten Algorithmen zur Vorhersage epileptischer Anfälle auch auf schaltungstechnischen Realisierungen von CNN, umgesetzt werden können. Ein Optimierungsverfahren für CNN-Parameter soll hierfür entwickelt und implementiert werden. Weiterhin ist die erforderliche Rechengenauigkeit auf CNN-Schaltungen von großer Bedeutung, da nur mit der Kenntnis der Anforderungen zukünftige schaltungstechnische Realisierungen entsprechend ausgelegt werden können.

Für die Untersuchungen stehen Daten von hirnelektrischen Ableitungen in Dateien zur Verfügung, welche für die Verwendung auf dem EyeRIS-System zuerst vorverarbeitet werden müssen. Hierbei entstehen durch die begrenzte Auflösung der AD/DA-Wandler erste Ungenauigkeiten, die in die Verarbeitung eingehen. Die Robustheit der Algorithmen zur Vorhersage von epileptischen Anfällen gegenüber geringer Auflösung der Wandler und möglicher schaltungstechnisch bedingter Effekte ist daher besonders wichtig. Die Ergebnisse sollen deshalb mit statistischen Methoden, der Receiver-Operating-Characteristic (ROC) und mittels speziell generierter Ersatzsignale – sogenannter Surrogate – genauer untersucht werden.

Im technischen Bereich soll die Anwendbarkeit auf die Problemstellung der Bildverarbeitung gelegt werden und die Möglichkeit von CNN basierten Algorithmen zur Erkennung von Prozessparametern bei Laserschweißverfahren untersucht werden. Ein solcher Prozessparameter ist das sogenannte Key-Hole, welches in Bildsequenzen von Laserschweißprozessen als ein Maß für die zu erwartende Qualität einer Schweißnaht herangezogen werden kann. Ein CNN basierter Algorithmus für die Erkennung solcher Key-Holes soll in dieser Arbeit vorgestellt und untersucht werden. Der CNN-Hardware-Teil des EyeRIS 1.1-Systems bietet hierfür die Möglichkeit, diesen direkt als



Kamera zu benutzen und somit sowohl die Bildaufnahme als auch die Verarbeitung schon auf dem EyeRIS 1.1-System vorzunehmen. Da die Regelung von Prozessen Anforderungen an die Rechenzeit von Steueralgorithmen stellen, soll hier die Rechengeschwindigkeit genauer beleuchtet werden.

In den folgenden Kapiteln wird zuerst eine allgemeine Einführung in die mathematische Beschreibung von CNN gegeben und die für diese Arbeit verwendete Plattform – das EyeRIS-System – genauer beschrieben, sowie die Ausführungsgeschwindigkeit der verfügbaren Operationen und die Darstellungsgenauigkeit der AD/DA-Wandler auf diesem System untersucht. Die Umsetzung eines Mustererkennungsverfahrens auf dem EyeRIS-System schließt sich an. Es folgt die Einführung eines neuen, speziell für dieses System entwickelten Optimierungsverfahrens. Dieses zeichnet sich durch Berücksichtigung von Symmetrien in gegebenen Problemstellungen besonders aus. Ergebnisse, die die Leistungsfähigkeit des Optimierungsverfahrens mit anderen Verfahren vergleichen, schließen sich an. Die für die folgenden Untersuchungen notwendigen mathematischen Auswerteverfahren werden ebenfalls erläutert und im speziellen auf die Receiver-Operating-Characteristic eingegangen. Im weiteren wird auf Anwendungen von CNN eingegangen. Hier soll beispielhaft die Anwendbarkeit in medizinischen und technischen Bereichen gezeigt werden. Zuerst wird die Vorhersage epileptischer Anfälle mittels zweier Verfahren unter Verwendung von CNNs untersucht, wobei auch der zuvor dargestellte Optimierungsalgorithmus zur Anwendung kommt. Danach wird ein Bildverarbeitungsalgorithmus für die Überwachung eines Laserschweißprozesses entwickelt und untersucht. Die jeweiligen Ergebnisse werden diskutiert und in der Zusammenfassung noch einmal gegenübergestellt. Im Anhang sind weitere Untersuchungsergebnisse dargestellt.



## 2 Grundlagen

Dieses Kapitel dient dazu, einen Überblick über Zellulare Systeme und im speziellen über die sogenannten Zellularen Neuronalen Netzwerke zu geben. Hierbei werden die einzelnen, ein CNN definierenden Eigenschaften wie Zustandsgleichung, Nachbarschaft der Zellen im Netz, Randbedingungen und andere vorgestellt und besprochen. Dieses Kapitel dient dazu, den Leser in die Lage zu versetzen die Anwendungen von CNN in späteren Kapiteln leichter zu erfassen und die verwendeten CNN-Typen einzuordnen.

### 2.1 Zellulare Systeme

In der Physik, Chemie und Biologie werden häufig Systeme aus miteinander wechselwirkenden Komponenten untersucht. Hierzu werden immer häufiger auch numerische Algorithmen im Rahmen von Computersimulationen verwendet. Zellulare Systeme, die der hohen Komplexität der Berechnungen durch eine massive parallele Verarbeitung Rechnung tragen, gewinnen zunehmend an Bedeutung. Ein solches System kann durch die folgende Definition 1 allgemein beschrieben werden.

**Definition 1:**

*Ein Zellulares System ist ein System mit räumlich unterscheidbaren Komponenten, die sich auf eine bestimmte, zu dem System gehörende, Art und Weise gegenseitig beeinflussen.*

Modell	Zustand	Zeit	Raum
partielle Differentialgleichungen	●	●	●
gekoppelte Gitter Differentialgleichungen CNN	●	●	○
gekoppelte Gitter Boltzmann Gittermodelle DT-CNN	●	○	○
interaktive Partikelsysteme	○	●	○
Gittergas Modelle Cellular Automata (CA)	○	○	○
○ = diskret ● = kontinuierlich			

**Tabelle 2.1:** Vergleich zwischen partiellen Differentialgleichungen, CNN, DT-CNN und CA bezüglich Zustand, Raum und Zeit (Klassifizierung gemäß [2]).

Hierbei können sowohl die Zustände als auch die zeitliche Veränderung des Systems kontinuierlich oder diskret sein. Hierdurch ergeben sich unterschiedliche Klassen zellulärer Systeme, die in Tabelle 2.1 zusammengestellt sind. Im weiteren sollen wichtige Klassen von CNN und zeitdiskreten CNN (DT-CNN) genauer beschrieben werden.

## 2.2 CNN

Zellulare Neuronale Netzwerke wurden im Jahre 1988 von L.O. Chua und L.Yang [1] eingeführt. Die Zustandsfunktion einer Zelle  $c_i$  eines Netzwerks ist gegeben durch

$$\frac{dx_i}{dt} = -x_i + \sum_{j \in \mathcal{N}_r(i)} A_j f(x_j) + \sum_{j \in \mathcal{N}_r(i)} B_j u_j + z_i \quad (2.1)$$

mit

1. Zelleingang (*Input*)  $u_{\mathbf{j}} \in \mathbb{R}$
2. Schwellwert der Zelle (*Bias*)  $z_{\mathbf{i}} \in \mathbb{R}$
3. Zellzustand  $x_{\mathbf{i}} \in \mathbb{R}$
4. Zellausgang (*Output*)  $y_{\mathbf{i}} \in \mathbb{R}$
5. Gewichtswerte  $\mathbf{A}_{\mathbf{j}}$  und  $\mathbf{B}_{\mathbf{j}}$ , welche die Gewichtungen von Zelle mit Indexvektor  $\mathbf{j}$  zu Zelle mit Indexvektor  $\mathbf{i}$  beschreiben und in den Gewichtsmatrizen  $\mathbf{A}$  und  $\mathbf{B}$  zusammengefaßt werden,

wobei der Zellausgang zumeist durch die stückweise lineare Funktion

$$y_{\mathbf{i}} = f(x_{\mathbf{i}}) = \frac{1}{2}(|x_{\mathbf{i}} + 1| - |x_{\mathbf{i}} - 1|) \quad (2.2)$$

gegeben ist, und die Position einer Zelle im Netzwerk durch den Koordinatenvektor  $\mathbf{i}$  bezeichnet wird. Die Zellzustandsgleichung (2.1) allein ist noch nicht ausreichend, um ein solches Netz zu beschreiben. Vielmehr wird zur vollständigen Beschreibung die Definition

- einer räumlichen Anordnung von Zellen (z.B. rechteckige, dreieckige oder hexagonale Gitterstruktur), das Netzwerk  $\mathcal{N}$ ,
- einer Nachbarschaft  $\mathcal{N}$  um eine Zelle, die durch den Radius  $r \in \mathbb{N}$  und ein Abstandsmaß  $\|\cdot\|$  festgelegt wird,
- einer Randbedingung bei räumlich begrenzten Netzwerken,
- sowie der Gewichte des Netzwerks in Matrixschreibweise.
- einer meist nichtlinearen Ausgangsfunktion z.B. (2.2), die abhängig vom Zustand den Zellausgang bestimmt,
- einer Zustandsfunktion z.B. (2.1), die basierend auf den Zuständen der Zelle selbst und der sich in der Nachbarschaft befindenden Zellen, die Zellzustände bestimmt,

benötigt. Betreffend der Gewichtungen werden zwei Netzwerktypen unterschieden, der translations invariante und der translations variante Typ. Ein translations invariantes Netzwerk kann für das gesamte Netzwerk durch Angabe von jeweils einer Gewichtsmatrix  $\mathbf{A}$  und

$\mathbf{B}$  die Netzwerkgewichtung festgelegt werden. Im Gegensatz dazu müssen für ein sogenanntes translations variantes Netzwerk für jede Zelle eigene Gewichtsmatrizen definiert werden, was einen erheblichen Mehrbedarf an Speicher für die Gewichte bedeutet. Dasselbe gilt auch für den Schwellwert  $z$ , ein translations invariantes Netzwerk benötigt hier nur die Angabe eines Wertes, für translations variante Netzwerke kann er von Zelle zu Zelle unterschiedlich sein. Die im Weiteren behandelten Netzwerktypen sind vom translations invarianten Typ.

### 2.2.1 Zellenanordnung

Die räumliche Anordnung der Zellen beeinflusst die Zustandsgleichung. Je nach der Anordnung sind unterschiedlich viele Nachbarn in ihr zu berücksichtigen. Wie schon einleitend erwähnt, kann man in einem zweidimensionalen Netzwerk rechteckige, dreieckige, hexagonale und andere Gitterstrukturen verwenden. Zumeist wird aber das Rechteckgitter gewählt, da dies sich am einfachsten realisieren läßt. Des Weiteren können mehrere Schichten solcher zweidimensionaler Netze miteinander verknüpft werden, so daß man zur Beschreibung der Position einer Zelle im Netzwerk zwei Koordinaten im zweidimensionalen Netzwerk und eine für die Schicht benötigt, in der sich die Zelle befindet. Allgemeiner kann hier von einem  $D$ -dimensionalen Raum ausgegangen werden, wobei mit einem Vektor mit Dimension  $D$  die Position von Zellen im Netzwerk beschrieben werden.

### 2.2.2 Nachbarschaft

Die Nachbarschaft einer Zelle  $c_{\mathbf{i}}$ ,  $\mathbf{i} \in \mathbb{N}^D$  des  $D$ -dimensionalen Netzwerks ist definiert als

$$\mathcal{N}_r(\mathbf{i}) = \{\mathbf{j} \mid \max(\|\mathbf{j} - \mathbf{i}\|) \leq r\}, \quad (2.3)$$

wobei  $r$  den Radius um die Zelle  $c_i$  und  $\|\mathbf{j} - \mathbf{i}\|$  eine Norm bezeichnet. Da den Zellen diskrete Koordinaten im gitterförmigen Netzwerk zugeordnet werden, kann man o.B.d.A.  $r \in \mathbb{N}$  annehmen. Im Falle einer Zellanordnung in einem rechteckigen Gitter, kann man Zellen als Quadrate auffassen, die vier Kanten und vier Ecken haben, an denen sie an andere Zellen des Netzwerks angrenzen. Für eine solche Anordnung können zwei wichtige Arten der Nachbarschaften angegeben werden [3] :

- die von-Neumann-Nachbarschaft, welche ausschließlich die an den Seiten einer Zelle angrenzenden Zellen zur Nachbarschaft zählt,

$$\mathcal{N}_r(\mathbf{i}) = \{\mathbf{j} \mid \sum_k |\mathbf{j}_k - \mathbf{i}_k| \leq r, r \in \mathbb{N}\} \quad (2.4)$$

wie in Bild 2.1(a) gezeigt. Die Anzahl der Zellen innerhalb dieser Nachbarschaft ist  $2r(r+1)+1$  also: 1, 5, 13, 25, 41, 61, ...

- die Moore-Nachbarschaft, welche zusätzlich auch die an den Ecken der Zellen angrenzenden Zellen mit hinzunimmt,

$$\mathcal{N}_r(\mathbf{i}) = \{\mathbf{j} \mid \|\mathbf{j} - \mathbf{i}\| \leq r, r \in \mathbb{N}\} \quad (2.5)$$

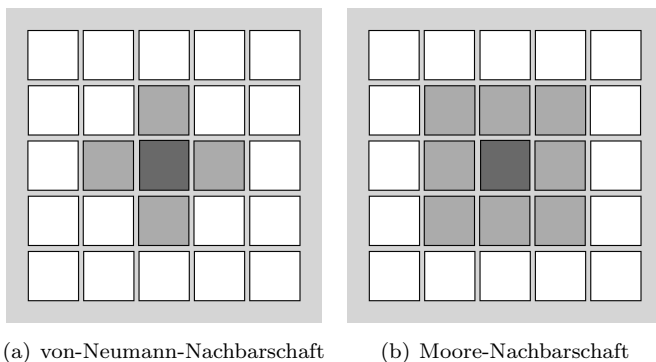
wie in Bild 2.1(b) gezeigt. Die Anzahl aller der Zellen innerhalb einer solchen Nachbarschaft ist  $(2r+1)^2$  also: 1, 9, 25, 49, 81, ...

Für ein einschichtiges Netzwerk ist dies für die Moore-Nachbarschaften in Bild 2.2 veranschaulicht.

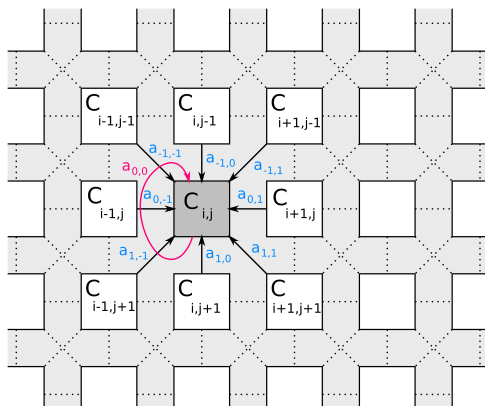
### 2.2.3 Randbedingungen

Die vorstehenden Betrachtungen und Definitionen lassen theoretisch unendlich ausgedehnte Netzwerke zu. In der Praxis jedoch ist ein Netzwerk  $\mathbf{N}$  zwangsläufig räumlich beschränkt, d.h. es existieren Randzellen  $c_{Rand}$ , für die nicht alle Nachbarzellen  $c_j, \mathbf{j} \in \mathcal{N}_r(\mathbf{i})$  Teil des Netzwerks sind:

$$\mathcal{B} = \{c_j \mid \mathbf{j} \in \mathcal{N}_r(\mathbf{i}) \wedge c_i \in \mathbf{N} \wedge c_j \notin \mathbf{N}\}. \quad (2.6)$$



**Bild 2.1:** Die zwei am häufigsten angewendeten Nachbarschaften (graue Zellen zählen zur Nachbarschaft).



**Bild 2.2:** Einfluß einer Kopplungs-Templates auf die Zelle in einem zweidimensionalen CNN (zweielementiger Indexvektor  $[i, j]$ ) mit direkter Moore-Nachbarschaft ( $r = 1$ ) vgl. Bild 2.1(b).



In praktischen Implementierungen werden diese Randzellen deshalb zum Netzwerk hinzugefügt [4], weshalb diese auch als virtuelle Zellen bezeichnet werden. Folgende Randbedingungen sind allgemeinen üblich:

### 1. Dirichlet-Randbedingungen

**Definition 2:**

*Die Zustände  $x_{c_j}$  der virtuellen Zellen  $c_j \in \mathcal{B}$  tragen mit einem konstanten Wert*

$$f(x_{c_j}) = \text{const} \quad , c_j \in \mathcal{B}.$$

zum Netzwerk bei.

### 2. von Neumann-Randbedingungen

**Definition 3:**

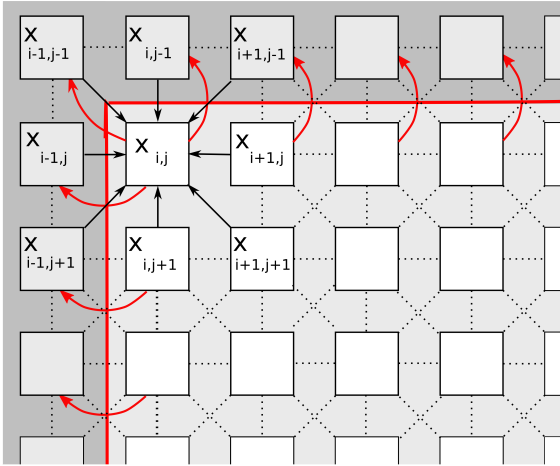
*Die Komponente der ersten Ableitung der Zustandsfunktion in Richtung der Normalen  $\vec{n}$  des Randes des Netzwerks ist Null*

$$\vec{n} \cdot \nabla f(x_{c_j}) = 0 \quad ,$$

womit die virtuellen Randzellen nicht zu Änderungen der Zellen im Netzwerk beitragen, was auch mit Zero-Flux bezeichnet wird. Hierdurch wird erreicht, daß die virtuellen Randzellen weder vom Netzwerk in irgendeiner Weise gespeist werden noch die Zellen des Netzwerks von den virtuellen Randzellen gespeist werden und so ein Gleichgewicht am Rand des Netzwerks hergestellt wird.

### Periodische Randbedingungen

Periodische Randbedingungen vermeiden die Einführung virtueller Randzellen, indem die Ränder des Netzwerks zusammengeführt werden. Dadurch entsteht im eindimensionalen Netzwerk ein geschlossener Ring, im zweidimensionalen ein Torus.



**Bild 2.3:** Virtuelle Randzellen bei Moore-Nachbarschaft in einem zweidimensionalen CNN (zweielementiger Indexvektor  $[i, j]$ ). Der Rand ist rot gekennzeichnet, die virtuellen Randzellen sind dunkelgrau unterlegt.

Wird wie im vorherigen Fall o.B.d.A. wieder von einem 1-dimensionalen Netz  $N$  mit  $n$  Zellen ausgegangen, so ergeben sich die absoluten Indizes der Zellen zu  $i \in \mathbb{Z}$  über die Modulo-Operation  $i \bmod n$ , d.h. ist  $i \in \mathbb{Z}$ , z.B.  $n + 1$ , wird auf die Zelle  $c_{i \bmod n} = c_1$  zugegriffen. In diesem Fall kann die Nachbarschaft der Zelle  $c_i$  mit  $i \in \{0, \dots, n\}$  auch gemäß

**Definition 4:**

$$\mathcal{N}_r(i) = \{j \mid (|j - i| \bmod (n - r)) \leq r\} \quad j \in \{0, \dots, n - 1\}$$

angegeben werden.

### 2.2.4 Templates

Damit das Netz verschiedene Operationen ausführen kann, sind noch Gewichtungen in den Beziehungen [5] zwischen den Zellen des Netzwerkes zu definieren. Diese gewichten den Einfluß der Zellen in der Nachbarschaft und können, wie im Folgenden für den Fall von zweidimensionalen Netzwerken, in Matrixschreibweise angegeben werden. Die Dimension der Matrix ist bei translations invarianten Netzwerken meist durch die symmetrisch gewählte Nachbarschaft vorgegeben. So definiert eine  $3 \times 3$  Nachbarschaft eine  $3 \times 3$  Gewichtsmatrix, die Template genannt wird und in ihrer allgemeinen Form gemäß

$$\mathbf{A} = \begin{pmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & a_{0,0} & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{pmatrix} \quad (2.7)$$

für die Feedforward-Template  $\mathbf{A}$  und die Feedback-Template  $\mathbf{B}$  dargestellt wird. Hierbei wird von der mittleren Zelle jeweils nach rechts bzw. links und nach oben und unten mit relativen Indizes auf die Gewichtswerte referenziert. Einige häufig verwendete Templates [5] sind

- **Average Template:** Diese Template führt eine Mittelung über die Zustände der Nachbarzellen aus. Der Zustand wird in Abhängigkeit von den Zellwerten in eine der möglichen Sättigungen (+1 oder -1) übergeführt:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad z = 0. \quad (2.8)$$

- Die **Threshold-Template:** ist der **Average-Template** ähnlich, nur daß sie nicht über Nachbarzellen mittelt und der angestrebte Sättigungswert nur von der Zelle selbst und dem Bias  $z$  abhängt. Zellen, die einen Wert größer als der Schwellwert

$z$  haben, werden zu  $+1$ , Zellen mit kleinerem Wert erreichen den Wert  $-1$ , Zellen, deren Wert gleich dem Schwellwert ist, sind in einem instabilen Gleichgewicht, was aber selten auftritt und in Hardwarerealisierungen von CNN infolge von Bauteilertoleranzen bzw. Rauschen nicht anzutreffen sein wird.

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad z = -p \quad (2.9)$$

mit

$$p \in [-1 \dots 1]. \quad (2.10)$$

- **Sobel-Operator-Template:** Für das Hervorheben von Kanten kann die Sobel-Operator-Template verwendet werden [6]. Linien oder Teile einer Linie in vertikaler Richtung werden durch

$$\mathbf{A}_v = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad z = 0 \quad (2.11)$$

hervorgehoben. Linien oder Teillinien in horizontaler Richtung werden durch

$$\mathbf{A}_h = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad z = 0 \quad (2.12)$$

hervorgehoben.

- **Small Pixel Removal:** Für die Beseitigung einzelner Pixel und auch zum Entrauschen von Bildern eignet sich die Small Pixel Removal Template

$$\mathbf{A} = \begin{pmatrix} 0 & 0.4 & 0 \\ 0.4 & 1.4 & 0.4 \\ 0 & 0.4 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 4.6 & -2.8 & 4.6 \\ -2.8 & 1 & -2.8 \\ 4.6 & -2.8 & 4.6 \end{pmatrix} \quad (2.13)$$

mit

$$z = -7.2 . \quad (2.14)$$

- **CCD-Template:** (Connected Component Detection Template) Diese Template verschiebt das gesamte Bild in eine von der Template bestimmte Richtung und reduziert dabei schwarze und weiße zusammenhängende Bereiche in einem Bild auf je eine Breite von einem Pixel. Dies erzeugt parallele Linien am Bildrand, deren Anzahl der Anzahl an SW-Übergängen im Ausgangsbild in der betrachteten Richtung, horizontal, vertikal oder diagonal, je nach CCD-Template, entspricht. Die Anzahl der resultierenden schwarzen und weißen Streifen gibt dann die Anzahl an SW-Übergängen an, wobei jeder Übergang einer zusammenhängenden Fläche im Ursprungsbild entspricht.

Als Beispiel sei die CCD-Template für die horizontale Richtung angegeben.

$$\mathbf{A}_h = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad z = 0 \quad (2.15)$$

Templates können klassifiziert werden, je nachdem, ob das Ergebnis ihrer Operation lokal begrenzt bleibt, oder Bildteile im Bild verschoben (propagiert) werden. Ferner unterscheidet man auch, ob sie bevorzugt auf Grauwertbilder oder auf schwarz/weiß-Bilder angewendet werden. Templates, deren Anwendung das Bild nicht verschieben, sind:

- Threshold-Template.
- Average-Template.
- Sobel-Operator-Template.
- Small-Pixel-Removal-Template.

Ein Beispiel für eine propagierende Template, die zugleich auch ein Beispiel einer Template darstellt, die nur auf schwarz/weiß-Bilder anzuwenden ist, ist die CCD-Template. Weitere Klassifizierungsmerkmale für Templates ergeben sich aus der Art des Netzwerktyps. Hierbei können die Einträge der Gewichtswerte in den Templatematrizen durch Funktionen oder Operatoren ersetzt werden.

### 2.2.5 Ausgangsfunktion

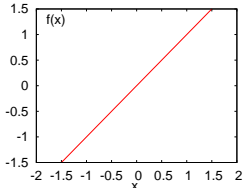
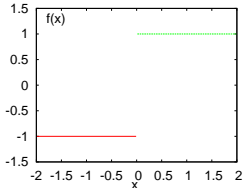
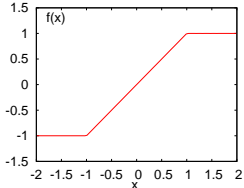
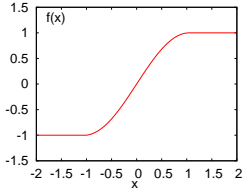
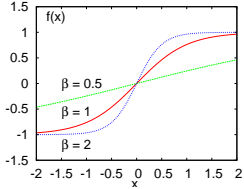
In der ursprünglichen Definition von Chua und Yang wurde gemäß (2.1) eine sogenannte stückweise lineare Funktion als Ausgangsfunktion gewählt. In den vergangenen Jahren wurden aber auch andere nichtlineare Ausgangsfunktionen eingeführt, von denen einige in Tab.2.2 aufgeführt sind.

### 2.2.6 Zustandsgleichung für verschiedene Klassen von CNN

In diesem Abschnitt wird auf spezielle Formen von CNN eingegangen, da das Verständnis einiger dieser Spezialformen für die späteren Untersuchungen wichtig ist. Dies gilt insbesondere für das zeitdiskrete CNN (DT-CNN) und das Full-Range Modell für die Hardwareimplementierung von CNN.

#### CNN mit ortsvariantem Bias

Eine andere Erweiterung des CNN ist der ortsvariante Bias  $z_i$ . Obwohl er für jede Zelle unterschiedlich ist, unterscheidet er sich von den Eingängen dadurch, daß er immer nur genau auf eine Zelle wirkt und über kein Template gewichtet und damit direkt eingespeist wird.

Funktion	$y_i$	Graph
Identität	$x_i$	
Sprungfunktion	$\Theta(x_i) \begin{cases} -1 & : x_{ij} < 0 \\ 1 & : x_{ij} \geq 0 \end{cases}$	
Stückweise lineare Funktion	$\frac{1}{2} \cdot ( x_i + 1  -  x_i - 1 )$ .	
$\sin(x_i \cdot \frac{\pi}{2})$ bis zur Sättigung	$\begin{cases} -1 & : x_i < -1 \\ \sin(x_i \cdot \frac{\pi}{2}) & : \text{sonst} \\ 1 & : x_i > 1 \end{cases}$	
Sigmoide Funktion	$\frac{2}{1+e^{-\beta x_i}} - 1 = \tanh \frac{\beta x_i}{2}$	

**Tabelle 2.2:** Häufig verwendete Ausgangsfunktionen von CNN.

## Das autonome CNN

Das autonome CNN zeichnet sich dadurch aus, daß es keine B-Template hat, also keinen äußeren Einflüssen unterliegt. Es ist nur auf sich selbst zurückgekoppelt, womit sich die Zustandsgleichung auf

$$\frac{dx_{\mathbf{i}}}{dt} = -x_{\mathbf{i}} + \sum_{\mathbf{j} \in \mathcal{N}_{\mathbf{r}}(\mathbf{i})} \mathbf{A}_{\mathbf{j}} f(x_{\mathbf{j}}) + z_{\mathbf{i}} \quad (2.16)$$

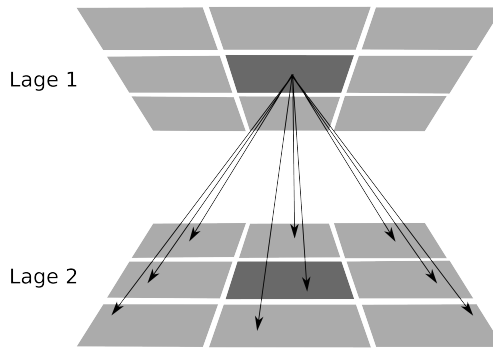
reduziert.

## CNN mit Verzögerung: Time-Delay-CNN (TD-CNN)

Zur Verallgemeinerung der Rückkopplungen innerhalb eines CNN kann man Verzögerungsglieder definieren, die die Ausgangswerte einzelner Zellen, oder der Eingänge, erst nach einer gewissen Zeitspanne (Delay) auf die Zellen zurückführen. Dies ermöglicht zum Beispiel die Erkennung von Bewegung in Bildern. Die Werte werden über sogenannte Delaytemplates gewichtet. Gl. (2.1) wird demzufolge zu

$$\begin{aligned} \frac{dx_{\mathbf{i}}}{dt} = & -x_{\mathbf{i}}(t) \\ & + \sum_{\mathbf{j} \in \mathcal{N}_{\mathbf{r}}(\mathbf{i})} \mathbf{A}_{\mathbf{j}} f(x_{\mathbf{j}}(t)) \\ & + \sum_{\mathbf{j} \in \mathcal{N}_{\mathbf{r}}(\mathbf{i})} \mathbf{A}'_{\mathbf{j}} f(x_{\mathbf{j}}(t - \tau_A)) \\ & + \sum_{\mathbf{j} \in \mathcal{N}_{\mathbf{r}}(\mathbf{i})} \mathbf{B}_{\mathbf{j}} u_{\mathbf{j}}(t) \\ & + \sum_{\mathbf{j} \in \mathcal{N}_{\mathbf{r}}(\mathbf{i})} \mathbf{B}'_{\mathbf{j}} u_{\mathbf{j}}(t - \tau_A) \\ & + z_{\mathbf{i}} . \end{aligned} \quad (2.17)$$





**Bild 2.4:** Mehrlagiges CNN; der Einflußbereich z.B. von Zelle 5 aus Lage 1 zu den Zellen der 2. Lage bei einer 3x3-Nachbarschaft ist kenntlich gemacht.

### Das zeitdiskrete Netzwerk (DT-CNN)

Ein Spezialfall des CNN ist das zeitdiskrete Netzwerk. Die Zustandsgleichung für zeitdiskrete CNN (DT-CNN) lautet analog zu (2.1):

$$x_i(t+1) = \sum_{j \in \mathcal{N}_r(i)} \mathbf{A}_j f(x_j(t)) + \sum_{j \in \mathcal{N}_r(i)} \mathbf{B}_j u_j(t) + z_i. \quad (2.18)$$

Zu bemerken ist hierbei, daß durch die zeitliche Diskretisierung [7] der Term  $-x_i$  entfällt.

### Mehrlagige CNN

Erweitert man das von Chua und Yang beschriebene Netzwerk um eine oder mehrere Zellschichten, im Folgenden als Lagen bezeichnet, so erhält man ein mehrlagiges CNN, wobei sich die Lagen  $m$  und  $m'$  gegenseitig beeinflussen. Gleichung (2.1) erweitert sich entsprechend

zu:

$$\begin{aligned} \frac{dx_{\mathbf{i}}^m}{dt} = & -x_{\mathbf{i}} + \sum_{m'} \left[ \sum_{\mathbf{j} \in \mathcal{N}_r(\mathbf{i})} \mathbf{A}_{\mathbf{j}}^{m'm} f(x_{\mathbf{j}}^{m'}) \right. \\ & \left. + \sum_{\mathbf{j} \in \mathcal{N}_r(\mathbf{i})} \mathbf{B}_{\mathbf{j}}^{m'm} u_{\mathbf{j}}^{m'} \right] \\ & + z_{\mathbf{i}}^m . \end{aligned} \quad (2.19)$$

Die Struktur eines solchen Netzes veranschaulicht Bild 2.4.

### Das Full-Signal-Range (FSR) Modell

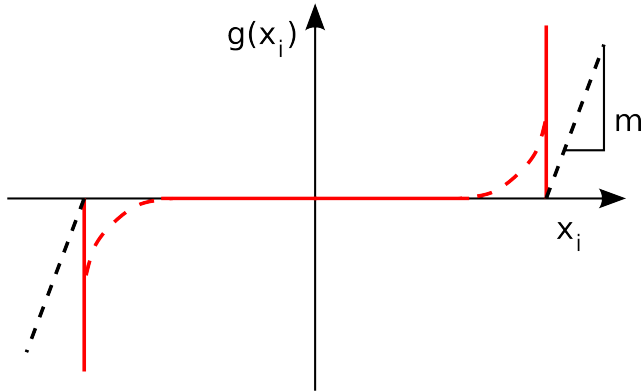
Aufgrund der nichtlinearen Ausgangskennlinie weist der Wertebereich des Ausgangs für die meisten Ausgangskennlinien eine Beschränkung von  $y_{\mathbf{j}}$  auf das Intervall  $[-1, 1]$  auf, doch der Zustand  $x_{\mathbf{i}}$  ist prinzipiell nicht beschränkt. Ein unbegrenzt steigender bzw. fallender Zustand läßt sich jedoch weder mit Strömen noch mit Spannungen schaltungstechnisch realisieren, weshalb das sogenannte Full-Range Modell

$$\frac{dx_{\mathbf{i}}}{dt} = -x_{\mathbf{i}} + \sum_{\mathbf{j} \in \mathcal{N}_r(\mathbf{i})} \mathbf{A}_{\mathbf{j}} f(x_{\mathbf{j}}) - g(x_{\mathbf{i}}) + \sum_{\mathbf{j} \in \mathcal{N}_r(\mathbf{i})} \mathbf{B}_{\mathbf{j}} u_{\mathbf{j}} + z , \quad (2.20)$$

speziell für schaltungstechnische Realisierungen vorgeschlagen wurde [8, 9, 10]. Hierbei kann der Term  $-x_{\mathbf{i}}$  durch Inkludierung in das Mittelelement der Gewichtsmatrix  $a'_{0,0} = a_{0,0} - 1$  in der Formel auch entfallen [11]. Das Full-Range Modell führt einen Verlustterm  $g(x)$  ein (Bild 2.5), der vom Zustand abhängt und folgendermaßen definiert ist:

$$g(x_{\mathbf{i}}) = \begin{cases} m(x_{\mathbf{i}} + 1) & : x_{\mathbf{i}} < -1 \\ 0 & : |x_{\mathbf{i}}| \leq 1 \\ m x_{\mathbf{i}} - 1 & : x_{\mathbf{i}} > 1 \end{cases} . \quad (2.21)$$

Die Steigung  $m$  im Verlustterm sollte hierbei so groß wie möglich gewählt werden. Im Idealfall sollte  $m = \infty$  gewählt werden, womit bei Überschreiten oder Unterschreiten des Wertebereiches  $[-1, 1]$  der Verlust  $\infty$  wird.



**Bild 2.5:** Verlustfunktion  $g(x_i)$  des Full-Range Modells. Rot gestrichelt ist hier eine für Schaltungen realistische Kurvenform an den Umsprungspunkten der Funktion gezeigt. Die Bedeutung von  $m$  in Formel 2.21 ist als Steigung der schwarz gestrichelten Linie dargestellt.

Hierdurch setzt eine Begrenzung des Zustands ein, wenn dieser sich der unteren bzw. oberen Grenze des, durch die Schaltung vorgegebenen, zulässigen Spannungsbereichs nähert. Schaltungstechnisch kann hierdurch eine Beschränkung auf die Betriebsspannung realisiert werden, wodurch sich allerdings die Dynamik eines solchen CNN ändert. Bei einem Wechsel des Vorzeichens der Summe aller Beiträge in der Zustandsgleichung reagiert ein CNN nach dem Full-Signal-Range Modell sofort mit einer Änderung seines Zustandes und seines Ausgangswertes  $f(x_i) = y_i$ , während bei einem CNN nach L.O.Chua und L.Yang [1]  $x_i$  in den linearen Bereich  $\pm 1$  der Ausgangsfunktion kommen muß, was bei einem Zustandswert  $|x_i| \gg 1$  eine Übergangs-

zeit benötigt und sich somit eine Änderung des Ausgangswertes erst bei Erreichen des linearen Bereichs der Ausgangsfunktion bemerkbar macht.

## 2.3 Numerische Simulation von CNN

Schaltungstechnische Realisierungen von CNN, wie unter anderem in [12, 13, 14, 15, 16, 17] beschrieben, sind auf spezielle Netzwerke oder Anwendungsbereiche zugeschnitten. Bei der Entwicklung neuer CNN-Algorithmen werden im allgemeinen Netzwerke betrachtet, für die noch keine schaltungstechnische Realisierung zur Verfügung steht oder deren Realisierung für diesen Zweck zu aufwändig wäre. Aus diesem Grund wird auch hier eine numerische Simulation von CNN auf Digitalrechnern durchgeführt.

Eines der ersten bedeutenden Simulationsprogramme ist SCNN [18, 19, 20], welches in der Arbeitsgruppe Komplexe Systeme am Institut für Angewandte Physik der J. W. Goethe Universität Frankfurt entwickelt wurde. Es besteht aus einem umfangreichen, monolithischen Programmkern, dessen Komponenten eng ineinandergreifen. Die Simulation und die Trainingsvorgänge werden durch eine Skriptsprache gesteuert [21], die eine automatisierte Bearbeitung großer Datenmengen zulässt. Die Komplexität und die nicht vorhandene deutliche Abgrenzung funktioneller Komponenten führt jedoch dazu, dass eine Erweiterung des Programms, wie etwa das Hinzufügen einer neuen Zellkopplungsvorschrift, nur aufwändig durchzuführen ist und daher Änderungen besonders fehleranfällig sind.

Aus diesen Gründen wurde in der Arbeitsgruppe [22, 23, 24] eine neue Simulationsumgebung, **F**lexible **O**bject **o**Riented **C**NN **E**nvironment -C++ library, kurz FORCE++ library, für CNN entwickelt,

welche einen objektorientierten Ansatz [25, 26] verfolgt. Bei FORCE++ handelt es sich um eine Klassenbibliothek, deren Klassen es ermöglichen, ein kompaktes Programm zu erstellen, welches nicht komplexer als ein vergleichbares Skript für SCNN ist. FORCE++ zeichnet sich durch folgende Eigenschaften aus:

- Objektorientiertheit; die dadurch bedingte Kapselung führt zu geringerer Fehleranfälligkeit bei Änderungen; jede Komponente kann getrennt getestet werden,
- leichte Erweiterbarkeit durch Ableiten von bereits vorhandenen Klassen, die allgemeine Schnittstellen zur Verfügung stellen,
- jede Komponente kann durch Modultests (unit tests) und automatische Testkonzepte [27] einzeln getestet werden,
- die Verwendung von Entwurfsmustern [28] führt zu klar strukturierten und gut verständlichen Codes.

Das Implementieren neuer Optimierungsverfahren gestaltet sich hierdurch für FORCE++ äußerst einfach. Das Optimierungsverfahren wird als abgeleitete Klasse eines verallgemeinerten Optimierungsverfahrens implementiert. Genauso kann ein beliebiges Fehlermaß implementiert und zur Anwendung gebracht werden. Neu implementierte Programmmodule bzw. Klassen können einfach mittels Modultests auf ihre Korrektheit überprüft werden.

Weiter unterstützt FORCE++:

- eigene Integrationsverfahren zu den bereits vorhandenen (Euler- und Runge-Kutta Verfahren) zu implementieren,
- frei programmierbare Zustandsgleichungen,
- schon jetzt 4 Optimierungsverfahren (Differential Evolution (DE), SSAPE, Broyden–Fletcher–Goldfarb–Shannon (BFGS), Simulated Annealing) [29, 30, 31]), die leicht um weitere ergänzt werden können,

- frei definierbare Fehlermaße für Optimierungsverfahren,
- variable Randbedingungen sowie die Erstellung eigener Randbedingungen,
- Symmetrien und variable Parameteranzahl sowie Definitionsbereiche von Parametern in Lernverfahren vorzugeben und polynomielle Gewichte,
- die Angabe von Trainingssets für Optimierungsverfahren.

All dies kann durch Implementieren und Einbinden eigener Programmteile, ohne die Klassenbibliothek verändern zu müssen, erreicht werden, was zur Stabilität von FORCE++ beiträgt. Dadurch kann man sich bei der Entwicklung auf die eigentliche Algorithmik konzentrieren und weniger mit spezifischen Aufgaben der Softwaretechnik befassen, was eine Erleichterung im Vergleich mit einer Erweiterung von SCNN bedeutet.

Im Zuge der vorliegenden Arbeit wurden unter anderem Simulationen mit reduzierter Rechengenauigkeit von 8 bit betrachtet. Während die Umsetzung dieser speziellen Anforderung mit dem monolithischen Programmkern von SCNN nur sehr schwer umzusetzen gewesen wäre und tiefgreifende Änderungen an vielen Stellen im Programm notwendig gemacht hätte, ist die Realisierung durch das objektorientierte, modulare Konzept von FORCE++ effizient zu lösen gewesen.

### 3 EyeRIS-System 1.1 mit analogem Ace16kv2 Focal-Plane Prozessor

Dank der sich rasch entwickelnden Halbleitertechnik ist es möglich, heute schaltungstechnische Realisierungen von CNN als integrierte Schaltkreise in Mikrochips herzustellen. Diese werden immer leistungsfähiger, was sich in der Anzahl der implementierten Zellen und deren Komplexität ausdrückt. Die leistungsfähigsten zur Zeit verfügbaren Systeme sind die auf Bildverarbeitung ausgerichteten EyeRIS-Systeme des Unternehmens Anafokus [14].

Das EyeRIS 1.1-System in Bild 3.1 kombiniert die Programmierbarkeit herkömmlicher digitaler Prozessoren (CPU <sup>1</sup>) mit einer analogen, hochgradig parallelen Verarbeitung auf Basis von zellularen Netzwerken [32]. Der digitale Prozessor wird als sogenannte Soft-Core-CPU durch ein Field-Programmable Gate Array (FPGA) realisiert, welches zu diesem Zweck als eine CPU mit dem NIOS II Kern des Herstellers Altera konfiguriert wurde. Dieser Prozessor dient der Kontrolle des analog arbeitenden ACE16kv2-Chips, der Nachverarbeitung von Daten und als Bindeglied zur Kommunikation mit einem PC. Durch die Anbindung an ein digitales System wie z.B. an einen PC kann das EyeRIS-System nicht nur im Stand-Alone Betrieb, sondern auch als Coprozessor benutzt werden. Der ACE16kv2 bietet durch seine hohe Parallelität bei einer Größe von nur ca. 1 cm<sup>2</sup> eine enorme Rechenleistung. Der digitale Teil des EyeRIS Systems

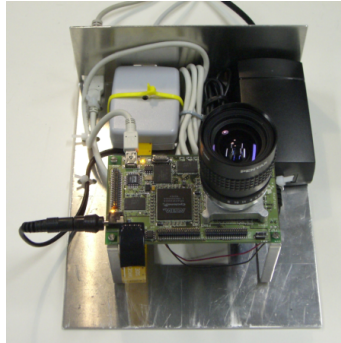
---

<sup>1</sup>Central Processing Unit

wird in der Programmiersprache C programmiert, für den analog arbeitenden Teil des Systems wird von der Herstellerfirma AnaFocus die Programmiersprache FPP, für Focal Plane Processing, bereitgestellt. Durch seine Fähigkeit, sowohl als Sensoranordnung zur Aufnahme von Bildern als auch zur parallelen Weiterverarbeitung zu dienen, bildet der ACE16kv2 gewissermaßen einen „intelligenten“ Sensor. Dank seiner Programmierbarkeit kann er für zahlreiche Aufgabenstellungen verwendet werden und insbesondere mit Hilfe eines Optimierungsverfahrens als adaptives System genutzt werden.



(a)



(b)

**Bild 3.1:** (a) EyeRIS-System ohne Optik (b) EyeRIS-System mit Optik und auf Halterung montiert.

### 3.1 ACE16kv2 und das Full-Range Modell

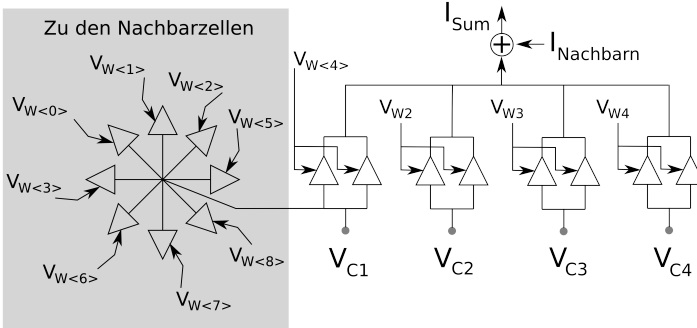
Der ACE16kv2 enthält 16384 Zellen die quadratisch in einem Netzwerk von 128 Zeilen und 128 Spalten angeordnet sind. Somit kann



jedes Pixel eines Bildes der Größe  $128 \times 128$  genau einer Zelle zugeordnet werden. Jede Zelle im ACE16kv2-Chip besitzt acht analoge Zustandsspeicher, LAM0 - LAM7, für Operanden. Wird ein Bild auf den ACE16kv2 geladen, so wird jedes Pixel genau einem Speicherbereich, z.B. LAM0, seiner korrespondierenden Zelle zugeordnet. Somit sind bis zu acht Datensätze als Bilder speicherbar, jedoch muß berücksichtigt werden, daß das Resultat einer Operation ebenfalls wieder in einen der acht Speicherbereiche abgelegt werden muß. Jede Zelle enthält einen sogenannten internen analogen Datenbus, den ACE-BUS[33]. Dieser ist durch eine einzige Leitung realisiert, da der Zustand zellintern durch genau ein analoges Signal realisiert wird. An diesen ACE-BUS sind die einzelnen Verarbeitungskomponenten über elektronische Schalter angebunden. Je nach gewünschter Operation werden die Verarbeitungskomponenten über die Schalter an den Bus angeschlossen und aktiviert. Für die Verarbeitung stehen vier Zwischenspeicher zur Verfügung, in welchen die Operanden einer Operation während deren Ausführung gespeichert werden. Diese Zwischenspeicher sind als Kondensatoren  $C_1, C_2, C_3$  und  $C_4$  realisiert, die vor der Ausführung einer Operation mit den jeweiligen Werten aus den Zustandsspeichern als Spannungen  $V_{x*}, * = 1 \dots 4$ , auf diese geladen werden. Die Gewichtswerte sind getrennt von den Zellen in einem Gewichte-Speicher abgelegt und werden allen Zellen gleichzeitig als Spannungen  $V_{W*}, * = 1 \dots 4$ , zugeführt. Hierbei dient  $*$  als Platzhalter für die vier verschiedenen Operanden und Gewichte. Die Multiplikation mit den Gewichten ergibt einen Strom, der mit den anderen Strömen mittels Stromsummation über einen Widerstand, die Summenbildung der Zustandsgleichung, wieder in eine Spannung gewandelt wird. Hieraus ergibt sich nach [12, 13, 34] die Zellzustandsgleichung für die Hardwarerealisierung von CNN im Focal Plane Processor ACE16kv2 zu:

$$\begin{aligned}
 V_{pixel} = & \sum_{i=0, i \neq 4}^8 V_{W1}[i]V_{x1}[i] + V_{Wx1}V_{x1}[4] & (3.1) \\
 & + V_{Wx2}V_{x2} + V_{Wx3}V_{x3} + V_{Wx4}V_{x4} ,
 \end{aligned}$$

wobei die  $V_{W_x}$  für die, die Gewichtswerte repräsentierenden Spannungen, die  $V_x$  für die Zellzustände repräsentierenden Spannungen stehen und  $V_{W_{x1}} \equiv V_{W1}[4]$  gilt, welches dem Mittelelement einer  $3 \times 3$  Template entspricht wie in Bild 3.2 illustriert.



**Bild 3.2:** Schematische Darstellung der Multipliziereinheit des EyeRIS-Systems. Die Gewichtung des Zentralelements  $V_w[4]$  und der Gewichtswerte  $V_{w2}$ ,  $V_{w3}$  und  $V_{w4}$  werden durch je zwei Multipliziereinheiten verrechnet und tragen somit doppelt zum Summenstrom  $I_{sum}$  bei. Es ist anzumerken, daß die Gewichtung der Nachbarbeiträge, aus schaltungstechnischen Gründen, nicht bei Eingang dieser, sondern bei der Ausgabe an die Nachbarzellen geschieht.

Die Multiplizierer liefern einen den Gewichten und den Eingangsspannungen proportionalen Strom  $I_{Sum}$  der sich mit

$$\begin{aligned} v_w &= V_{w0} - V_{w*} \\ v_x &= V_{x*} - V_{x0} \end{aligned} \quad (3.2)$$

zu

$$\begin{aligned} I_{Sum} &= k \left\{ \sum_{i=0, i \neq 4}^8 v_{W1}[i] v_{x1}[i] + 2v_{W_{x1}}[4] v_{x1}[4] \right. \\ &\quad \left. + 2v_{W_{x2}} v_{x2} + 2v_{W_{x3}} v_{x3} + 2v_{W_{x4}} v_{x4} \right\} \end{aligned} \quad (3.3)$$

ergibt [12]. Hierbei sind  $V_{w0}$  und  $V_{x0}$  schaltungstechnisch bedingte vorgegebene Spannungen und  $k$  eine Konstante gegeben durch

$$k = 1.2510^{-6} A/V^2.$$

Der Faktor 2 kommt durch die interne Realisierung der Multiplikatoren, wie in Bild 3.2 abgebildet, zustande. Hier ist gezeigt, wie die Signale  $V_{C_i}$  jeweils zwei Multiplizierern zugeführt werden, welche zu einem doppelten Strom am Ausgang und einer doppelten kapazitiven Belastung der Eingangsquellen führt. Die Gewichte zu den Nachbarzellen tragen jeweils nur einfach bei. Der Ausgangsstrom der Multiplizierer wird der Ausgangsfunktion zugeführt. In Abhängigkeit der gewünschten Ausgangsfunktion werden zwei sogenannte Limiting Modes unterschieden[12]. Im ACE16kv2-Chip wird der Soft-Limiting Mode durch die Implementierung einer Verlustfunktion

$$I_g = g(v_x) = \begin{cases} +\infty & : x_i > 1 \\ 0 & : |x_i| \leq 1 \\ -\infty & : x_i < -1 \end{cases}, \quad (3.4)$$

realisiert, wodurch der Zustand einer Zelle durch

$$C \frac{dv_x}{dt} = -I_g + I_{Sum} = -g(v_x) + I_{Sum} \quad (3.5)$$

gegeben ist. Im Hard-Limiting Mode ergibt sich der Zellzustand durch

$$C \frac{dv_{out}}{dt} = I_{Sum}. \quad (3.6)$$

Je nach Anwendung wird einer der zwei Betriebsarten des Chip gewählt, der Hard-Limiting Mode, wenn die Schwellwertfunktion angewendet werden soll, bzw. Soft-Limiting Mode, wenn der Verlustterm des Full-Range Modelles vor dem Speichern zur Anwendung kommen soll.

Wie aus der Gl.(3.1) hervorgeht, bietet der ACE16kv2-Chip die Möglichkeit, vier Werte pro Zelle miteinander zu verrechnen, was als ein Netzwerk mit vier CNN-Lagen, wie in 2.4 für zwei Lagen gezeigt, aufgefaßt werden kann. Allerdings ist nur ein Wert mit einer  $3 \times 3$  Template und den korrespondierenden Werten der Nachbarzellen verrechenbar, die weiteren Gewichte sind skalare Größen, so daß die anderen Werte nur den Zellwert selber beeinflussen und nicht die Nachbarzellen. Die drei skalaren Gewichtswerte tragen wie das Mittelelement der  $3 \times 3$  Template, schaltungstechnisch bedingt, mit dem Faktor 2 doppelt zum Ausgangsstrom der Multiplizierer bei, d.h. der Gewichtswert wird schaltungsintern bereits doppelt gewertet, weshalb Templates für die Verwendung auf dem Chipsystem entsprechend angepaßt werden müssen. Eine Rückkopplung findet nur im Soft-Limiting Modus statt [35] und dient dazu, eine zu schnelle Zustandsentwicklung in die Sättigung zu verhindern. Hierbei wird bei Durchführung der Convolution-Operation (ein Befehl der FPP-Sprache) eine der drei skalaren Kopplungsgewichte und nicht die  $3 \times 3$  Gewichtungsmatrix mit einem stark negativen Gewicht vorgelegt, womit in dieser Betriebsart effektiv nur zwei skalare Werte und Werte aus der Nachbarschaft der einzigen  $3 \times 3$  Gewichtsmatrix miteinander verrechnet werden können. Durch die eingeschränkte Wahl einer Rückkopplung im Soft-Limiting Mode sind Templates aus [5] nur mit Anpassungen oder gar nicht für die Anwendung auf dem ACE16kv2 geeignet. Im Falle des Hard-Limiting Modes sind sogar nur solche Templates, die keine Rückkopplungen benötigen, ohne Einschränkungen anwendbar. Bei propagierenden Templates, wie z.B. der CCD-Template, wirkt sich dies dahingehend aus, daß diese nicht mehr propagieren und ihre Funktionsweise verloren geht. Um in solch einem Fall ein propagierendes Verhalten in der Verarbeitung zu erreichen, muß dies durch iterative Anwendung einer solchen Template, wie z.B. der CCD-Template, in einer Schleife erfolgen.

### 3.1.1 Bildverarbeitung mit dem ACE16kv2

Die Datenverarbeitung mit dem ACE16kv2 ist vielseitig, weshalb hier die mit dem ACE16kv2 zu realisierenden Operationen noch einmal genauer unter dem Gesichtspunkt der Bildverarbeitung betrachtet werden, wobei die zu verarbeitenden Daten als Bilder vorliegen.

Faßt man die Zustände als die Werte der Pixel eines Bildes, repräsentiert durch Spannungen, auf und berücksichtigt, daß vier Bilder in einer Operation miteinander verrechnet werden können, so kann man die Pixelwerte einer Zelle und ihrer Umgebung mit  $\mathbf{P}_A, P_B, P_C$  und  $P_D$  bezeichnen, wobei im Folgenden  $P_D = z$  den Schwellwert repräsentiert und im Weiteren als translations invariant aufgefaßt wird. Die Ausgangswerte der Multiplizierer werden durch Ströme repräsentiert und zu  $I_{Sum}$  aufsummiert. Der Strom aus den Multiplizierern des ACE16kv2 ergibt sich dann zu [10]

$$I_{Sum} = \mathbf{A} \bullet \mathbf{P}_A + b \cdot P_B + c \cdot P_C + z, \quad (3.7)$$

wobei  $\bullet$  die Faltung einer  $3 \times 3$  Parametermatrix, also die Anwendung der  $3 \times 3$  Template

$$\mathbf{A} = \begin{pmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & a_{0,0} & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{pmatrix} \quad (3.8)$$

ist, sowie  $b$  und  $c$  einzelne Gewichtswerte und  $z$  den Schwellwert bezeichnen (vergleiche hierzu Gl.(3.3)). Die Matrix

$$\mathbf{P}_A = \begin{pmatrix} P_{A_{-1,-1}} & P_{A_{-1,0}} & P_{A_{-1,1}} \\ P_{A_{0,-1}} & P_{A_{0,0}} & P_{A_{0,1}} \\ P_{A_{1,-1}} & P_{A_{1,0}} & P_{A_{1,1}} \end{pmatrix} \quad (3.9)$$

ist hierbei ein Bildausschnitt um die Zelle, auf den die Template angewandt wird. Folgende Anwendungsfälle kann man unterscheiden [10, 12]:

- Es sei  $\mathbf{A}$  eine  $3 \times 3$  eine Template (siehe Abschnitt 2.2.4), sowie  $c = z = 0$  und  $b = -1$  die weiteren Parameter der auf  $\mathbf{P}_A$  anzuwendenden Operation. Der in jeder Zelle das Signal repräsentierende Strom wird an den Kondensator  $C_B$  der jeweiligen Zelle geleitet; dann ergibt sich

$$C_B \frac{dP_B}{dt} = -P_B + \mathbf{A} \bullet \mathbf{P}_A \quad (3.10)$$

für die Zustandsgleichung. Der stabile Endzustand ist dann als  $P_B = \mathbf{A} \bullet \mathbf{P}_A$  gegeben.

- Wird  $C_A$  anstelle von  $C_B$  verwendet, so ergibt sich

$$C_A \frac{dP_A}{dt} = -P_A + \mathbf{A} \bullet \mathbf{P}_A, \quad (3.11)$$

wodurch eine dynamische Kopplung zwischen den Zellen in der Nachbarschaft, die sich gegenseitig beeinflussen, ergibt, was einem CNN-artigen Verhalten entspricht. Diese Konfiguration bedingt allerdings, daß zur Stabilisierung im Soft-Limiting Mode die Rückkopplung des Zentralelements der Gewichtstemplate einen großen negativen Wert hat. Dies bedeutet, daß eine solche Rückkopplung einer Zelle auf sich selbst nicht im Sinne eines CNN aufzufassen ist. Im Hard-Limiting Mode gibt es keine Rückkopplung.

- Wird  $C_A$  verwendet und das Mittelelement von  $\mathbf{A}$  auf  $-1$ , alle anderen auf 0 gesetzt, dann folgt für den stabilen Endzustand [10]

$$P_A = b \cdot P_B + c \cdot P_C + z \quad . \quad (3.12)$$

Dies entspricht der Anwendung von arithmetischen Operationen auf Grauwertbildern.

### 3.1.2 Ausführungsgeschwindigkeit einzelner Routinen

Auf dem EyeRIS-System kommen zwei Prozessoren zum Einsatz, der analog arbeitende ACE16kv2-Prozessor und ein digitaler Prozessor. Zur Programmierung des digitalen Teils des EyeRIS-Systems wird C bzw. C++ verwendet, für den analogen ACE16kv2 die von Anafocus entwickelte Programmiersprache FPP. Der FPP-Code wird hierbei in den C-Code in spezielle, durch Präcompiler Direktiven geschützte Bereiche, mit eingebettet. Der FPP-Code selbst besteht aus sogenannten Sektionen, die durch die C-Funktion *ExecuteSection(NameDerSektion)* aufgerufen und ausgeführt werden können. Für die Entwicklung und die Laufzeitoptimierung von Algorithmen für das EyeRIS-System ist es wichtig, die Ausführungsgeschwindigkeiten der einzelnen Operationen sowohl auf dem analogen als auch auf dem digitalen Teil, unter Berücksichtigung der Latenz der Kommunikationswege der zwei Teilsysteme des EyeRIS-Systems, zu kennen. Zu diesem Zweck wurden Ausführungsgeschwindigkeiten der unterschiedlichen Instruktionen gemessen und in den Tabellen 3.1 und 3.2 zusammengefaßt, um einen Überblick über das Laufzeitverhalten zu erhalten. Diese können zum Abschätzen des Laufzeitverhaltens von CNN-Algorithmen auf dem EyeRIS-System verwendet werden, ersetzen aber nicht die Messung des Laufzeitverhaltens eines komplexen Algorithmus.

Befehl	Zeit ( $\mu s$ )	Bemerkung
C for-Schleife	0.16	
C TimeMeasure	0.44	0.42 ohne return value
C ExecuteSection	0.97	Zeit im FPP-Code eine Sektion aufzurufen
C-Funktionsaufruf	0.17	Zeit eines Funktionsaufrufs mit gleichen Parametern wie ExecuteSection

**Tabelle 3.1:** Zeitmessung für C-Routinen.

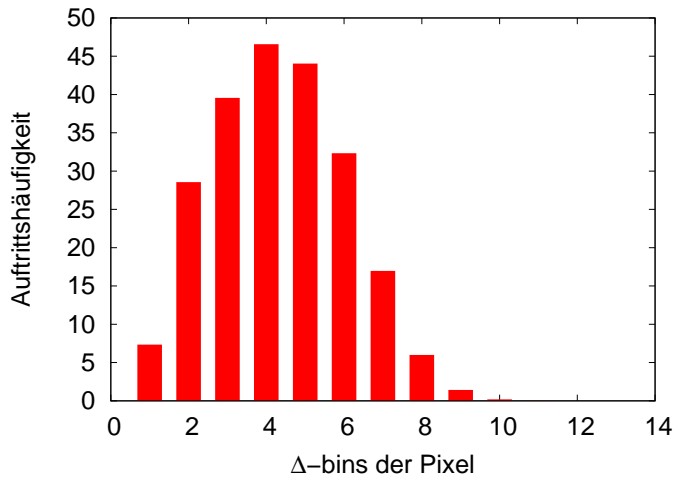


Befehl	Zeit ( $\mu s$ )	Bemerkung
FPP set	0.1	
FPP inc / dec	0.1	
FPP comp	0.15	
FPP inc + comp	0.35	
FPP jump	0.07	
FPP Convolution (binary)	14.7	
FPP Convolution (gray)	18.7	
FPP TemplateExecute hard	15.7	
FPP TemplateExecute soft	19.7	
FPP InitializeFPP	2.7	
FPP LoadVinInLam	1.7	in C for-Schleife
FPP LoadVinInLam	0.86	in FPP Schleife aus comp inc und jump
FPP LoadImageToFPP	188.26	in C for-Schleife
FPP DownloadImageFromFPP	178.68	in C for-Schleife
FPP InternalImageTransfer	1.7	in C for-Schleife
FPP InternalImageTransfer	1.998	FPP loop aus comp inc und jump
FPP LogicOperation ein Operand	4.085	
FPP LogicOperation zwei Operanden	6.072	
FPP DownloadPartialImage	4.3 - 4.7	per Gruppe á 4 Zeilen im Mittel $4.6\mu s$
FPP LoadAdressEvent	0.27 - 0.29	per Pixel gemessen mit schwarzem Bild
FPP LoadAdressEvent	0.7 - 0.8	per Pixel gemessen mit weißem Bild
FPP InitOpticalAquisition	2.45 - 2.56	
FPP ReadOpticalBuffer	73.7	
FPP OpticalAquisition	932.7	timebase ms
FPP OpticalAquisition	4.0	timebase $\mu s$
FPP OpticalAquisition	3.7	timebase ns

**Tabelle 3.2:** Zeitmessung für FPP-Routinen.

## 3.2 Darstellungsgenauigkeit von Grauwerten

Zur Bestimmung der Darstellungsgenauigkeit von Zellzuständen wurden Untersuchungen bezüglich der Grauwertdiskriminierung des analog arbeitenden ACE16kv2-Chips durchgeführt. Hierzu wurden alle  $2^8 = 256$  mit 8-bit darstellbaren Grauwerte als Schwellwert auf 256 Bilder mit jeweils einheitlichem Grauwert aus dem Bereich  $0 - 255$  mittels der Threshold-Template angewandt. Auf jedes dieser Bilder mit gegebenem Grauwert wurden somit unterschiedliche Schwellwerte zur Grauwertdiskriminierung angewendet. Hierbei unterscheidet sich der Schwellwert vom jeweiligen Grauwert des Bildes um jeweils  $\Delta = \pm 0, \pm 1, \pm 2 \dots \pm 14$ . An den Wertebereichsgrenzen (0 bzw. 255) wurde der Bereich der zu untersuchenden Schwellwerte entsprechend eingegrenzt, um einen Zahlenbereichsüberlauf zu vermeiden, z.B. kann man bei einem Grauwert des Bildes von 250, Schwellwerte nur bis maximal +5 untersuchen, da bei  $250 + 6 = 256$  man bereits eine in 8-bit nicht mehr darstellbare Zahl erhält. Es wurde jeweils ausgezählt, wie viele Pixel richtig bzw. falsch durch den jeweiligen Schwellwert diskriminiert werden konnten. Es zeigte sich, daß die Anzahl abweichender Pixel vom Sollgrauwert abhängt. Für die Auswertung wurde eine Auszählung abweichender Pixel für gegebenen Grauwert und gegebenem *Delta* des Schwellwertes in sogenannten  $\Delta$ -Bins durchgeführt. Jeder  $\Delta$ -Bin repräsentiert hierbei die Abweichungshäufigkeit eines untersuchten Grauwertunterschieds des jeweiligen Bildes zum jeweils angewendeten Schwellwert. Wird ein einheitlich graues Bild in den ACE16kv2 geladen, so wird der Grauwert durch schaltungstechnisch bedingte Abweichungen in den Zellen mit einer Spannung repräsentiert, die mit einer wie in 3.3 gezeigten Verteilung nach Rückwandlung in einen digitalen Wert einem anderem Grauwert entspricht. Die Auswertung und Zusammenfassung mit Normierung unter Berücksichtigung der Anzahl Pixel von  $128 \times 128$  im jeweiligen Bild und der Anzahl untersuchter Grauwerte (256 Grau-



**Bild 3.3:** Zusammengefaßte und auf die Pixelanzahl normierte Statistik der Abweichungshäufigkeit in Abhängigkeit von den untersuchten Grauwertdifferenzen  $\Delta$ -Bins. In einem Bild mit gegebenem Grauwert weicht die Representation dieses Grauwertes durch die Zellzustände bei den meisten Zellen vom Sollgrauwert um  $\pm 4$  davon ab, die maximale Abweichung vom Sollgrauwert ist  $\pm 13$ .

wertstufen) ergab, daß eine Diskriminierung erst bei  $\pm 13$  mit Sicherheit durchzuführen ist, allerdings die Mehrzahl der Pixel bereits bei einem Grauwertunterschied von  $\pm 4$ , was 8 Grauwertstufen entspricht, gut getrennt werden können (siehe Bild 3.3). Hierbei stehen die Werte der  $\Delta$ -Bins für die Häufigkeit von Abweichungen vom Sollergebnis. Die Ergebnisse zeigen, daß, obwohl die AD/DA-Wandler eine Auflösung von 8 bit haben, im allgemeinen mit Abweichungen zu rechnen ist, die einer Reduzierung der Wertebereichsauflösung um 3 bit auf nur 5 bit entspricht. Dies leitet sich aus der Abweichung von 8 Grauwertstufen ab, die sich mit 3 bit – da  $8 = 2^3$  ist – darstellen lassen. Dies ist allerdings nur für die hier am häufigsten auftretende Abweichung gültig, im extremsten gefundenen Fall von  $\pm 13$  ist diese Abweichung wesentlich größer. Da bei realen Anwendungen in den wenigsten Fällen mit einheitlich grauen Bildern gearbeitet wird und auch Leckströme im Chip zu ähnlichen Abweichungen führen können, ist von Mischfehlern auszugehen, weshalb die hier gefundene Auflösungsreduzierung nur als grobes Maß für die obere Grenze der möglichen Auflösung betrachtet werden kann. Weiterhin kann es durch die Anwendung verschiedener Templates dazu kommen, daß sich mögliche Fehler in der Grauwertdarstellung verstärken oder auch wieder ausgleichen können. Weitere Fehler können noch durch die Nichtidealität der AD/DA-Wandler verursacht werden, weshalb man eine allgemein gültige Fehlergröße kaum angeben, sondern die jeweilige Anwendung berücksichtigen und sofern möglich robust gegen solche Fehler machen muß.

## 4 Musterdetektion mit dem ACE16kv2

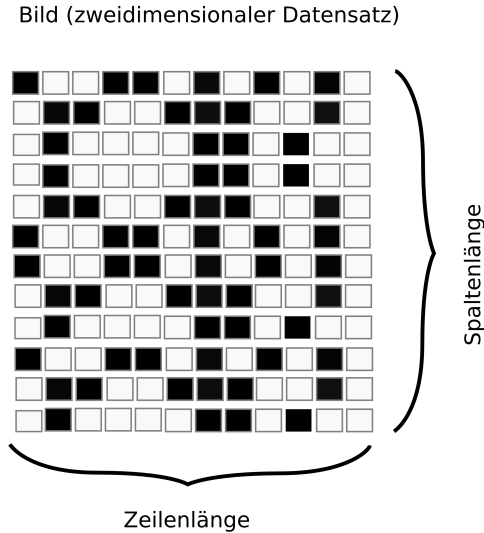
Die im Folgenden vorgestellte Musterdetektion arbeitet auf zwei-dimensionalen Datensätzen, Bildern, wobei einzelne Pixel  $p$  hierbei durch die euklidischen Koordinaten  $(p_x, p_y)$  referenziert werden. Für die Mustererkennung mit CNN bietet sich die PatternMatching-Finder-Template [5]

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} \quad z = -N+0.5 \quad (4.1)$$

an, wobei die Definition

$$b_{i,j} = \begin{cases} 1 & \text{wenn Pixel im zu erkennenden Muster schwarz ist} \\ 0 & \text{wenn Pixel im zu erkennenden Muster nicht relevant ist} \\ -1 & \text{wenn Pixel im zu erkennenden Muster weiß ist} \end{cases} \quad (4.2)$$

gilt und  $N$  die Anzahl der in  $\mathbf{B}$  nicht zu Null gesetzten Einträge angibt. Durch die Struktur der Template bedingt, definiert ein  $3 \times 3 = 9$  Pixel großer Bereich jeweils ein Muster, was in Bild 4.1 für eine bestimmte Form der Datenanordnung veranschaulicht wird. Jedes dieser Pixel kann entweder schwarz oder weiß sein, womit sich  $2^9 = 512$  unterschiedliche Muster ergeben. Man kann die Muster, durch Aneinanderreihung der Musterzeilen, auch als 9-stellige Binärzahl auffassen und jedem Muster die korrespondierende Dezimalzahl zuordnen, was die Referenzierung vereinfacht. Zum Beispiel entspricht das binäre Muster  $010101010_b$  (beachte das  $b$  im Index für binär) der



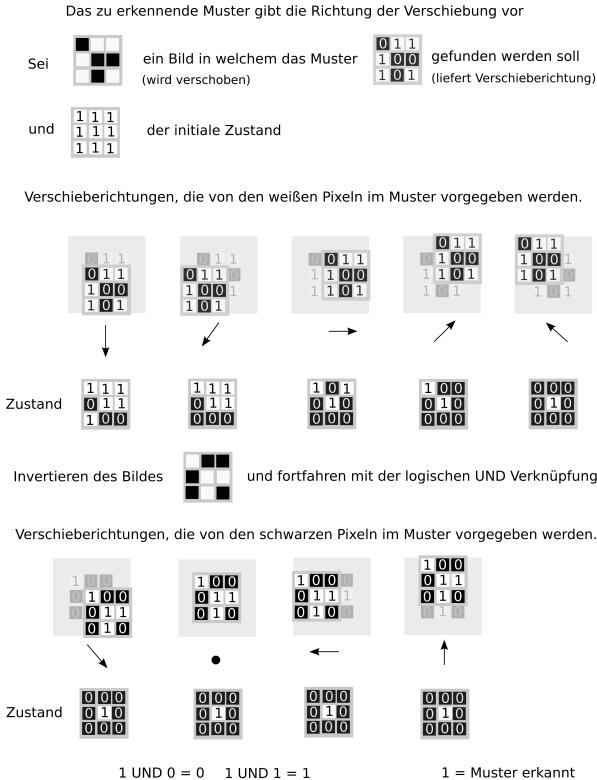
**Bild 4.1:** Ein Bild als 2D-Datensatz.

Dezimalzahl  $170_d$  (beachte das  $d$  im Index für dezimal) welches in Bild 4.6 dargestellt ist. Die Bestimmung der Musterhäufigkeiten kann in den Ergebnisbildern durch einen digitalen Algorithmus sehr effizient durchgeführt werden. Auf dem EyeRIS-System kann durch die AdressEvent-Funktion direkt die Anzahl weißer bzw. – nach Invertierung – die vormals schwarzen Pixel ausgelesen werden. Diese Funktion liefert die Koordinaten von maximal 4096 weißen Pixel im Bild, wobei die Zählung in der oberen linken Ecke bei Pixel  $(0 : 0)$  beginnt. Sollten mehr als 4096 weiße Pixel im Bild vorhanden sein, so werden die überzähligen Pixel in Richtung der Koordinaten  $128 : 128$  nicht mehr berücksichtigt. Voruntersuchungen auf dem EyeRIS-System zeigten, daß die PatternMatchingFinder-Template auf dem EyeRIS-System nicht zur Anwendung kommen konnte, da das EyeRIS-System, wie in Kapitel 3 beschrieben, keine

---

Rückkopplung im Sinne eines CNN besitzt. Für die Mustererkennung auf dem EyeRIS System muß daher ein anderer Algorithmus entwickelt werden. Der im Folgenden vorgestellte PatternMatching-Algorithmus basiert auf der logischen Verknüpfung von Pixeln in Kombination mit Verschiebe-Operationen und Invertierung des Bildes. Die Pixel des zu untersuchenden Bildes werden den Zellen des ACE16kv2 zugeordnet, was die Bildgröße in diesem Fall auf  $128 \times 128$  beschränkt.

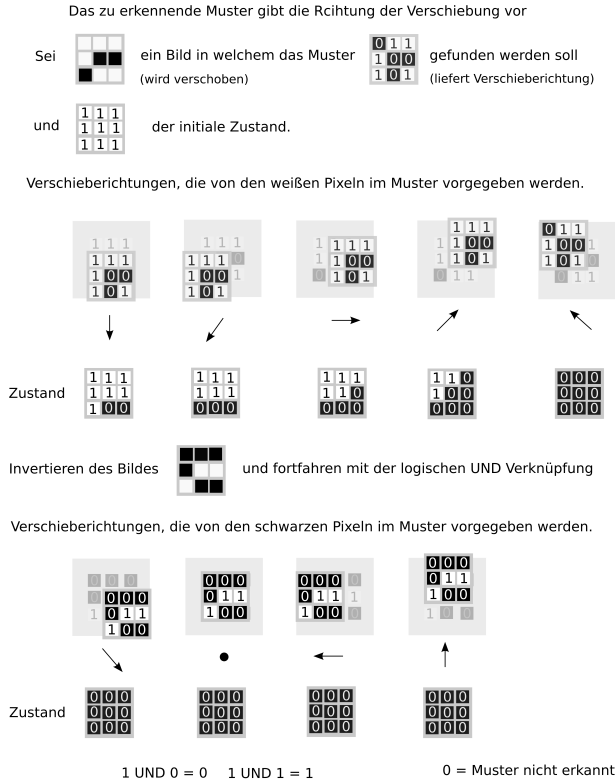
Das Ergebnisbild, welches die Ergebnisse der im folgenden auszuführenden logischen UND Verknüpfungen kumuliert, wird mit weiß, logisch „wahr“, initialisiert. Die im folgenden verwendeten Operationen, Verschieben und logische UND-Verknüpfung, werden auf allen Pixelkoordinaten  $(p_x, p_y)$  parallel ausgeführt. Für jedes Pixel mit Koordinate  $(p_x, p_y)$  im Ergebnisbild wird im Eingangsbild die Nachbarschaft des Pixels mit korrespondierenden Koordinaten untersucht. Hierbei zählt das Pixel selber auch zu seiner Nachbarschaft (2.2.2). Zuerst werden Nachbarpixel im Eingabebild, welche im zu findenden Muster weiß, logisch „wahr“, sind, nacheinander an die Koordinate  $(p_x, p_y)$  des Pixels im Ergebnisbild verschoben und mit dem Pixelwert an der Koordinate des Pixels im Ergebnisbild logisch UND verknüpft. Das Ergebnis einer solchen UND-Verknüpfung wird im Ergebnisbild an der Koordinate  $(p_x, p_y)$  gespeichert und in weiteren UND-Verknüpfungen weiterverwendet, siehe Bild 4.2. Sollte eine solche Operation zu einem „unwahr“ führen, da im Eingabebild an der Stelle, an der im Muster ein weißes Pixel erwartet wurde, aber im Eingabebild tatsächlich ein schwarzes Pixel vorhanden ist, so ändert sich der Wert des Pixels an der Koordinate  $(p_x, p_y)$  in „unwahr“, das Muster wurde also an der Stelle  $(p_x, p_y)$  nicht gefunden, wie in Bild 4.3 ab Schritt 4 gezeigt ist. Nach Abarbeitung aller Pixel in der Nachbarschaft des Pixels an der Koordinate  $(p_x, p_y)$  die im zu findenden Muster weiß sind, wird das Eingabebild invertiert. Anschließend werden alle jene Pixel, die im zu findenden Muster schwarz sein sollen, wie zuvor durch Verschieben und logische UND-Verknüpfung untersucht. Durch die Invertierung sind nun schwarze Pixel weiß und



**Bild 4.2:** Shift- und UND-Operationen zur Erkennung eines im Bild enthaltenen Musters. Grau dargestellt ist die ursprüngliche Position der Pixel im Bild. Pfeile zeigen in Richtung der Verschiebung.

weiße Pixel schwarz. Sollte also an der Stelle im zu findenden Muster ein schwarzes Pixel sein, so sollte dieser im Eingabebild, wenn das Muster an der Koordinate  $(p_x, p_y)$  zu finden sein sollte, weiß sein. Die logische UND-Verknüpfung, sofern im Ergebnisbild nicht schon ein schwarzes Pixel („unwahr“) vorhanden ist, sollte ein „wahr“ liefern,





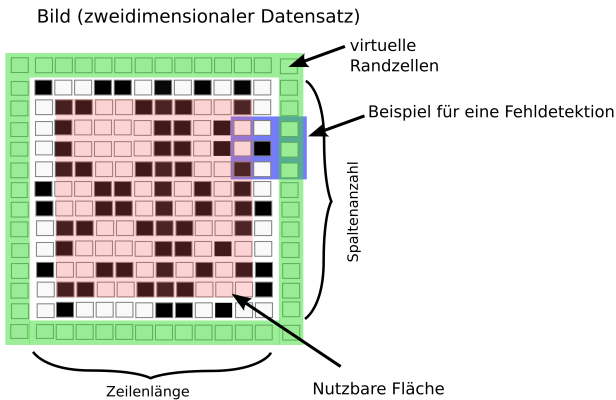
**Bild 4.3:** Shift- und UND-Operationen zur Erkennung eines nicht im Bild enthaltenen Musters. Grau dargestellt ist die ursprüngliche Position der Pixel im Bild. Pfeile zeigen in Richtung der Verschiebung.

also ein weißes Pixel im Ergebnisbild.

Wurden alle Pixel der Nachbarschaft geprüft, ist im Ergebnisbild an den Pixelkoordinaten, an denen das zu findende Muster vorhanden ist, ein weißes Pixel, ansonsten ein schwarzes Pixel. Sollen schwar-

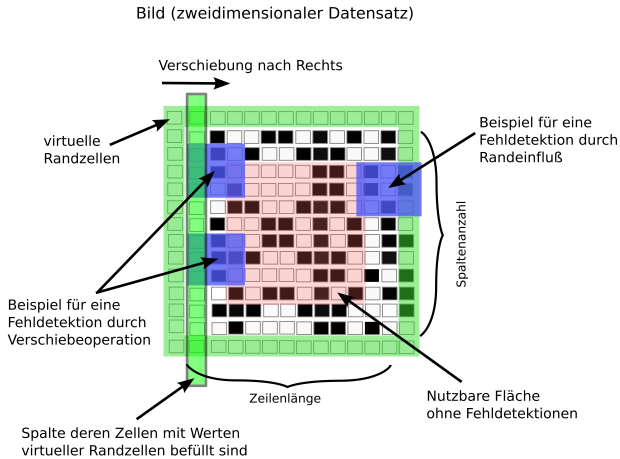
ze Pixel zur Kenntlichmachung der Pixelkoordinaten, an denen das Muster gefunden wurde, verwendet werden, so muß das Ergebnisbild invertiert werden.

Durch die Verschiebeoperationen und den Einfluß von virtuellen Randzellen ist nicht die gesamte Fläche für die fehlerfreie Detektion von Mustern geeignet. Der Einfluß der virtuellen Randzellen führt bei den Zellen, die zum Netz gehören aber virtuelle Randzellen zur Nachbarschaft haben, unter Umständen zur Detektion von Mustern, die nicht im ursprünglichen Bild enthalten sind, wie in Bild 4.4 gezeigt wird. Durch die Verschiebeoperationen des Mustererkennungsalgorithmus



**Bild 4.4:** Mögliche Detektion von Mustern die nicht im Bild enthalten sind.

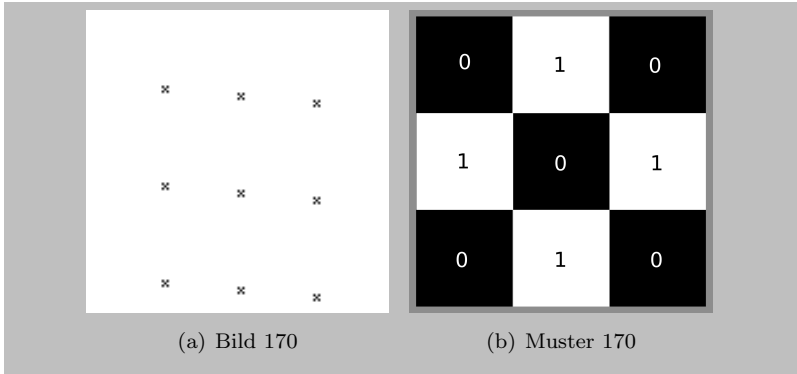
werden Zellwerte der virtuellen Randzellen an den Kanten des Netzwerks in dieses hineinverschoben. Dies führt zu einer Ausweitung des für eine fehlerfreie Detektion nicht nutzbaren Bereiches des Netzwerkes, wie in Bild 4.5 gezeigt ist. Dies muß bei der Anwendung des Algorithmus beachtet und bei der Auszählung von erkannten Mustern mit berücksichtigt werden.



**Bild 4.5:** Mögliche Detektionsfehler durch in das Netz Hineinverschieben von Werten aus den virtuellen Randzellen. Dies führt zu einer weiteren Reduzierung der nutzbaren Fläche für fehlerfreie Mustererkennung.

## 4.1 Implementierung auf dem EyeRIS System

Die Implementierung des beschriebenen Verfahrens wird für das EyeRIS System im Folgenden genauer untersucht. Hierfür wird für jedes mögliche der 512  $3 \times 3$ -Muster, ein Bild erzeugt. In jedem dieser Bilder wurde das jeweilige Muster an neuen Stellen auf dem  $128 \times 128$  großen weißen Bild, wie es Bild 4.6(a) für das Muster 170 4.6(b) zeigt, eingebettet. Zum Vergleich wird der Algorithmus mit Verschiebe-Operationen und logischen Operationen auf einem PC implementiert. Dieser wird seinerseits mit einem Musterdetektionsalgorithmus, der auf direktem Vergleich von Pixelwerten und Musterpixelwerten auf einem PC basiert, verglichen.



**Bild 4.6:** (a) Das an verschiedenen Stellen im  $128 \times 128$  großen Bild eingebettete Muster 170. (b) Das  $3 \times 3$  Muster 170 in vergrößerter Darstellung. Die Bezeichnung ergibt sich aus der binären Darstellung der Zahl  $170 = 010101010$  mit schwarz= 0 und weiß= 1 und Anordnung in als  $3 \times 3$  Matrix.

## 4.2 Ergebnisse

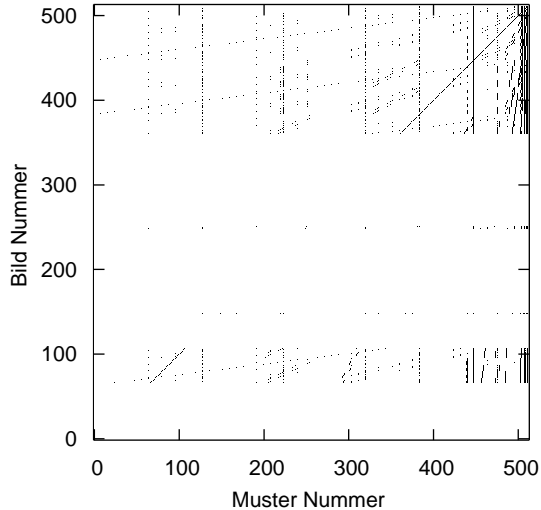
Die in der Programmiersprache C++ implementierten Referenzalgorithmen lieferten für jedes Bild und Muster übereinstimmende Ergebnisse, die Implementation auf dem verwendeten Chip-System hingegen wies Abweichungen auf. Hierbei wichen die fehlerhaften Ergebnisbilder von den Referenzbildern überwiegend durch eine Verschiebung nach links vom Sollbild ab, was im Folgenden noch genauer betrachtet wird.

Es wurde die Musterdetektion auf dem verwendeten Chip-System mit allen 512 möglichen  $3 \times 3$  Mustern auf alle 512 Bilder, die wie zuvor beschrieben durch die neunfache Einbettung jeweils eines Musters erzeugt wurden, angewandt, um so die Verlässlichkeit der Detektion mit dem EyeRIS-System zu untersuchen. Hierbei wurden die beiden möglichen Iterationsreihenfolgen untersucht:

1. erst alle Muster nacheinander in einem Bild suchen, dann nächstes Bild laden etc.,
2. erst ein Muster in allen Bildern suchen, dann nächstes Muster in allen Bildern etc.

Es wurden  $512^2 = 262144$  Musterdetektionen für jede der zwei möglichen Anwendungsreihenfolgen durchgeführt. Im ersten Fall (alle Muster über ein Bild dann nächstes Bild) waren sehr viele Fehldetektionen zu beobachten. Die Untersuchung der Ergebnisbilder ergab, daß die Rechtsverschiebung auf dem ACE16kv2 im Detektionsalgorithmus in den meisten fehlerhaften Ergebnisbildern nicht ausgeführt wurde und dieses fehlerhafte Verhalten in nicht systematischen Intervallen auftritt. In der überwiegenden Anzahl der Muster und Bilder funktionierte jedoch die Verschiebung. Aufgrund der langen Laufzeit und der Notwendigkeit das EyeRIS-System ab und an neu starten zu müssen, wurden die Untersuchungen in mehreren Etappen durchgeführt. Es wurden Bilder in folgenden Blöcken 0 – 14, 15, 16 – 60, 61 – 106, 107 – 148, 149 – 168, 169 – 206, 207 – 251, 252 – 333 und 334–511 in jeweils einem Durchlauf verarbeitet. In Bild 4.7 sind Fehldetektionen, die entweder auf die Detektion eines nicht vorhandenen Musters oder die Nichtdetektion eines vorhandenen Musters hindeuten, als schwarze Punkte kenntlich gemacht. Das EyeRIS-System liefert für diese Reihenfolge der Mustersuche deutlich hohe Anzahl von Fehldetektionen. Die weißen Bereiche im Bild zeigen fehlerfreie Musterdetektionen an. In den anderen Bereichen sind deutlich Linien zu erkennen, die auf eine systematische Fehldetektion hinweisen. Wenn im Chip-System erst einmal eine Fehldetektion auftritt, bleibt das Chip-System bis zum Neuinitialisieren eines weiteren Bildblocks und Neustart in einem solchen fehlerhaften Betriebsmodus. Genauere Analysen der Ergebnisse deuten darauf hin, daß der systematische Fehler auf eine Kombination von Shift-Fehlern in der Musterdetektion und einem „Einbrenneffekt“ des jeweiligen Bildes zurückzuführen ist. Dieser Effekt läßt sich durch die zweite genannte Reihenfolge der Verarbeitungsschritte oder durch Laden eines weißen Bildes zwischen

je zwei Musterdetektionen vermeiden.



**Bild 4.7:** Muster-zu-Musterbild-Fehler. Jeder schwarze Punkt zeigt eine Muster-zu-Musterbild-Kombination, in der ein Fehler aufgetreten ist. Es wurden die Bilder in folgenden Blöcken 0 – 14, 15, 16 – 60, 61 – 106, 107 – 148, 149 – 168, 169 – 206, 207 – 251, 252 – 333 und 334 – 511 in jeweils einem Durchlauf verarbeitet. Die untersuchten Bilder haben alle den Aufbau wie in 4.6 gezeigt, nur mit verschiedenen eingebetteten Mustern.

Im zweiten Fall (Muster über alle Bilder dann nächstes Muster) sind bei Mustern, die viele weiße Pixel enthalten, Fehler (siehe Bild 4.8) zu erkennen, die nicht direkt vom untersuchten Muster, sondern von einzelnen fehlerhaften Zellen des Chips und dem Rand herrühren.

Hierbei repräsentieren dunklere Pixel Zellen mit größeren Abweichungen vom Sollwert. Pixel am Rand des CNN-Netzwerks sind in

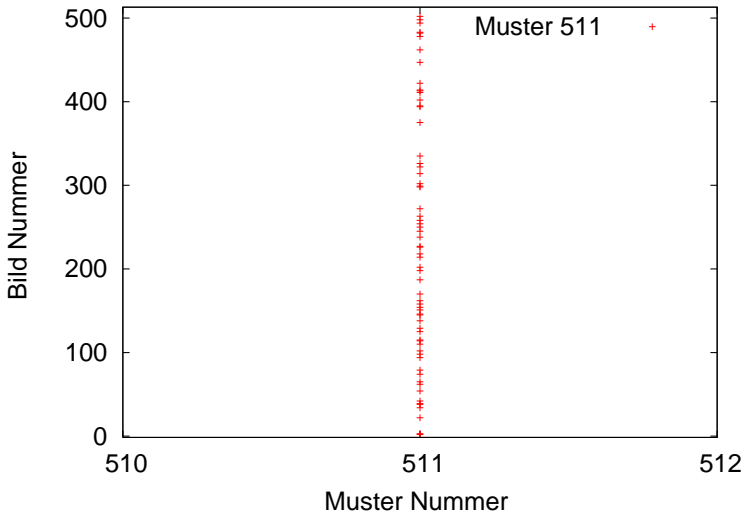


**Bild 4.8:**  $128 \times 128$  Bild mit deutlich erkennbaren grauen bis schwarzen Pixeln, welche korrespondierende fehlerhaft arbeitende Zellen markieren. Deutlich zeichnet sich ab, daß Pixel am Rand oftmals fehlerhaft sind.

den überwiegenden Fällen fehlerhaft. Deshalb wurden diese bei der Bewertung, ob ein fehlerhaftes Bild vorliegt, nicht berücksichtigt. Fehlerhafte Detektionsergebnisse treten hierbei unter Nichtberücksichtigung der Randzellen nur im Fall eines komplett weißen Musters gemäß Bild 4.9 auf.

Die gewonnenen Ergebnisse lassen erkennen, daß die Musterdetektion als grundlegende Komponente des hier gezeigten Verfahrens auf der zur Untersuchung verwendeten schaltungstechnischen Realisierung prinzipiell funktioniert, eine eindeutige und gänzlich fehlerfreie Musterdetektion jedoch nicht möglich ist.

Aufgrund möglicher Fehler in der Musterdetektion bei Verwendung der schaltungstechnischen Realisierung ist es nicht zuverlässig möglich, das Auftreten oder Nichtauftreten eines einzelnen Musters oder einer Musterkombination zur Prädiktion von epileptischen Anfällen heran-



**Bild 4.9:** Muster-zu-Musterbild-Fehler. Die Verarbeitung erfolgt indem für ein gegebenes Muster aus den 512 möglichen Mustern dieses in allen 512 Bildern mit eingebetteten Mustern gesucht wurde und erst dann mit dem nächste Muster aus 512 Mustern in der gleichen Weise fortgefahren wurde.

zuziehen. Daher wurden die Musterhäufigkeiten in einer statistischen Auswertung untersucht. Aufgrund der Anordnung der Daten in einem Bild und der Definition der Muster als  $3 \times 3$  Matrix, bot es sich an, eine Auszählung der in einem Muster vorkommenden SW-Übergänge, wie in Abschnitt 7.1.5 näher erläutert wird, sowohl in horizontaler als auch in vertikaler Richtung zur Einteilung in Musterklassen durchzuführen.



## 5 Parameteroptimierung

In den vergangenen Jahren wurden unterschiedliche Optimierungsverfahren für die Verwendung mit CNN angepaßt bzw. neu entwickelt [21, 36, 37, 38]. In der Regel wurden hierfür Simulationsplattformen verwendet, deren Rechengenauigkeit und Darstellungsgenauigkeit von Parametern im Bereich der Gleitkommazahlen liegt. Schaltungstechnische Implementierungen von CNN verlangen jedoch besondere Parameteroptimierungen nicht zuletzt, um vorhandene Parametersätze und CNN-Algorithmen an die gegebene Hardware anzupassen. Auch wenn CNN schaltungstechnisch rein digital implementiert werden können, so spiegeln solche Schaltungen eher Simulationsalgorithmen wider und implementieren simulationstechnisch zeitaufwändige Teile eines Algorithmus in Schaltungsstrukturen. Die analoge schaltungstechnische Realisierung von CNN kommt der ursprünglichen Definition der CNN als werte- und zeitkontinuierliche Systeme näher. Allerdings ist zur Anbindung an digitale Schaltungen und Rechner eine AD/DA Wandlung nötig. Die Implementierung von Optimierungsverfahren für Hardwarerealisierungen der CNN muß die begrenzte und diskrete Auflösungsgenauigkeit der in schaltungstechnischen Realisierungen von CNN verwendeten AD/DA-Wandler berücksichtigen. Weiterhin sind unter anderem hardwarebedingte spezifische Parameter, wie z.B. der Spannungsbereich zur Darstellung von Parametern oder des Zustandes einer Zelle zu berücksichtigen und gegebenenfalls mit in die Optimierung einzubeziehen.

Für Optimierungsverfahren ist die Vorgabe eines Fehlermaßes notwendig. Hierdurch wird bestimmt, wie gut ein durch das Optimierungsverfahren bestimmter Parametersatz die gestellte Problemstel-

lung erfüllt. Die Menge aller möglichen Parametersätze zusammen mit zugehörigen Fehlern bildet einen Parameterraum. Ein Optimierungsverfahren sucht nun diesen Parameterraum nach Minima ab, um den optimalen Parameterstz zu finden. Optimierungsverfahren können in verschiedene Klassen gegliedert werden:

1. solche, die eine analytische Berechnung der Parameter zulassen, wenn das gegebene zu optimierende Problem mathematisch berechenbar ist,
2. solche, die Information über den Gradienten des Fehlers im Parameterraum verwenden,
3. Verfahren, die eine Berechnung des Gradienten nicht benötigen,
4. jene, die theoretisch das globale Extremum erreichen, also die für die Problemstellung optimale Lösung finden,
5. Optimierungsverfahren, die nur lokale Extrema finden, welche zwar Lösungen der Problemstellung darstellen, obwohl u.U. bessere Lösungen existieren.

Da der Gradient im allgemeinen nicht immer als bekannt bzw. herleitbar vorausgesetzt werden kann, ist es von Interesse, auch Verfahren näher zu untersuchen, welche diese Information nicht benötigen. Beispiele für Lernverfahren, die keine Gradienteninformation benötigen und darüber hinaus theoretisch in der Lage sind, das globale Extremum zu finden, sind Optimierungsverfahren, welche

- den physikalischen Prozeß des langsamen Abkühlens von Schmelzen (Simulated Annealing) nachbilden und
- evolutionär inspirierte Optimierungsverfahren, welche den Vorgang der biologischen Evolution, wie sie Charles Darwin (1809 - 1882) [39] aufstellte, nachbilden.

Letztere, auch als künstliche Evolutionsverfahren [40] bezeichnete Verfahren, müssen nicht alle Einzelheiten des biologischen Vorbildes exakt nachbilden, um als Optimierungsverfahren zu funktionieren.

Die biologische Evolution besteht im wesentlichen aus den Schritten Vermehrung, Mutation/Cross-Over und Selektion, welche sich auch bei der künstlichen Evolution wiederfinden. Das Problem, eine Optimierung auf CNN-Hardware unter der Berücksichtigung von einschränkenden, durch die Problemstellung gegebenen Randbedingungen zu implementieren, kann eine hohe Komplexität erreichen. Gradientenbasierte Verfahren sind mitunter nicht anwendbar, wenn etwa die eindeutige Bestimmung von Gradienten aus den Zuständen der Zellen nicht mehr möglich ist. In solchen Fällen wird häufig auf heuristische Verfahren, wie das auf physikalischen Betrachtungen basierende Simulated Annealing oder das aus der Biologie stammende Modell der simulierten Evolution, zurückgegriffen.

## 5.1 Fehlermaß

Für die folgenden Betrachtungen muß, um die Güte einer Optimierung objektiv vergleichbar zu machen, auf die Verwendung einheitlicher Fehlermaße geachtet werden. Für die in dieser Arbeit durchgeführten Untersuchungen an zweidimensionalen CNN wurden die folgenden Fehlermaße verwendet:

- Der Relative Root Mean Square Error (RRMSE)

$$RRMSE = \sqrt{\frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N \frac{(x_{i,j} - y_{i,j})^2}{\widehat{y}^2}}, \quad (5.1)$$

wobei  $M \cdot N$  die Anzahl aller Werte,  $x_{i,j}$  die Istwerte,  $y_{i,j}$  die Sollwerte und  $\widehat{y}^2$  der Mittelwert der Sollwertquadrate ist.

- Der Mean Square Error (MSE)

$$MSE = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N (x_{i,j} - y_{i,j})^2 \quad (5.2)$$

mit  $M \cdot N$  der Gesamtzahl der summierten Werte,  $x_{i,j}$  der Istwert und  $y_{i,j}$  der Sollwert für eine Zelle an der Position  $i, j$  in einem zweidimensionalen Netzwerk.

Diese Fehlermaße zeichnen sich zum einen durch ihre Einfachheit aus und sind somit auch auf dem EyeRIS-System noch relativ effizient zu berechnen. Zum anderen bietet sich insbesondere der RRMSE an, da in anderen Untersuchungen ebenfalls dieses Fehlermaß häufig verwendet wird und somit ein Vergleich der Ergebnisse möglich ist. Für die Implementierung auf dem EyeRIS-System mit seinen  $128 \times 128$  Zellen sei weiterhin erwähnt, daß der MSE aufgrund des durch die AD/DA-Wandlung beschränkten Wertebereiches von 256 Stufen (8 bit) eine Obergrenze nicht überschreiten kann. Diese Obergrenze leitet sich aus dem MSE her, wobei  $N \cdot M = 128 \times 128$  die Gesamtzahl der Zellen im Netzwerk,  $x_{i,j}$  den jeweiligen Zellausgangswert und  $y_{i,j}$  den Referenzwert an der Position  $i, j$  im Referenzbild bezeichnet, weshalb der maximale MSE für die vorgegebene Auflösungsgenauigkeit von 8 bit durch

$$MSE_{max} = \frac{1}{128^2} \sum_{i=1}^{128} \sum_{j=1}^{128} 255^2_{i,j} = 255^2 . \quad (5.3)$$

gegeben ist.

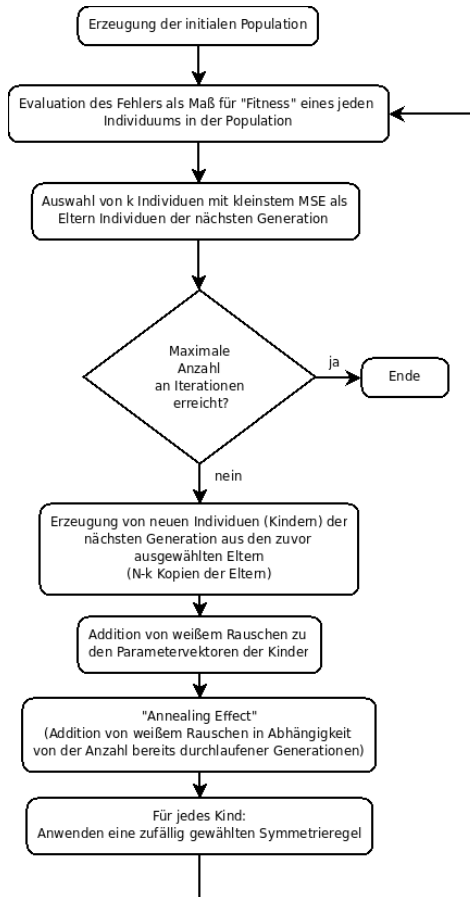
## 5.2 Optimierungsverfahren für CNN

Zur Optimierung von Parametersätzen bei CNN sind die lokalen Kopplungen und die Translationsinvarianz von Templates vorteilhaft, da sich somit die Anzahl der zu optimierenden Parameter reduziert und von der Anzahl der Zellen im Netzwerk unabhängig ist. Des weiteren sind oftmals die Problemstellungen derart, daß Symmetrien

in den Problemstellungen vorhanden sind, auch wenn diese nicht unbedingt von vornherein bekannt sind. Der Symmetry Supporting Annealed Parameter Evolution (SSAPE) Optimierungsalgorithmus ist ein evolutionärer Optimierungsalgorithmus, der mit dem Simulated Annealing kombiniert wurde, um so eine bessere Stabilität der Endergebnisse zu gewährleisten und Symmetrien in der Problemstellung bzw. eine implizite aber unbekannte Symmetrie in der Lösungsmenge während der Optimierung mit berücksichtigt. Bild 5.1 auf Seite 56 zeigt den Flußgraphen des SSAPE-Algorithmus.

Der SSAPE-Algorithmus wurde in Hinblick auf eine gewünschte Robustheit gegenüber Bauteiltoleranzen unter Berücksichtigung der räumlichen CNN-Struktur speziell für das EyeRIS-System entwickelt, aber auch in das Simulationssystem FORCE++ integriert. Ein Parametervektor kann hierbei zum einen aus den Gewichtswerten der Zellen und zum anderen aus Parametern, welche die Funktionsweise der elektrischen Schaltungen der Zellen bestimmen, zusammengesetzt sein. Da der SSAPE-Algorithmus evolutionär inspiriert ist, wird ein Parametervektor im Folgenden auch als Individuum  $I$  bezeichnet. Werden  $N$  Individuen zu einer Menge zusammengefaßt, so spricht man auch von einer Population  $P$  der Mächtigkeit  $N$ . Betrachtet man diese Population zu einem Zeitpunkt  $t$ , so wird diese Menge auch Generation  $G_t$  zum Zeitpunkt  $t$  genannt. In jedem Iterationsschritt wird eine Untermenge von  $k$  Individuen (Eltern) aus der Gesamtmenge aller Individuen einer Generation in Abhängigkeit vom Fehler  $E_i$  des jeweiligen Parametersatzes  $i$  ausgewählt.

Das Fehlermaß dient hierbei als sogenannte Fitnessfunktion, so daß der genannte Prozess das Prinzip des „Survival of the Fittest“ der Evolution nachbildet. Die Individuen der folgenden Generation  $G_{t+1}$  werden aus den zuvor ausgesuchten  $k$  Individuen der Generation  $G_t$  zum Zeitpunkt  $t$  gebildet, indem diese vervielfältigt und deren



**Bild 5.1:** Flußgraph des SSAPE-Algorithmus. In jedem Iterations-schritt werden  $k$  Individuen mit kleinstem Fehler der Gesamtmenge aller Individuen der aktuellen Generation ausgewählt. Diese bilden die Eltern der nächsten Generation. Neue Parametervektoren (Kinder) werden als Kopien der Eltern angelegt und durch Rauschen und Anwenden von zufällig gewählten symmetrieerzeugenden Regeln mutiert.

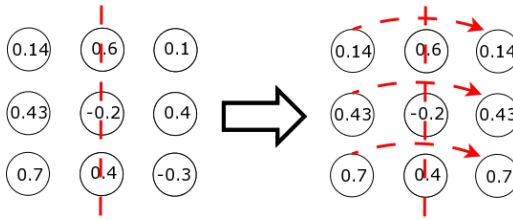
Parameterwerte eines jeweiligen Individuums mit Zufallszahlen aus dem Zahlenbereich  $[-1, 1]$

$$A_{Grundmutation} \in [-1, 1] \quad (5.4)$$

gewichtet werden, bis wieder  $N$  Individuen gebildet wurden. Eine zweite Gewichtung kann durch Anwenden von Zufallszahlengewichten aus dem Bereich  $[-1/j, 1/j]$ , die zur aktuellen Iterationstiefe  $j$  invers proportional ist,

$$A_{AnnealedMutation} \sim 1/j \quad (5.5)$$

geschehen. Dieser Einfluß wird bei jeder folgenden Iteration geringer, was einem „Auskühleffekt“, einem Annealing, entspricht. Die Zufallszahlen sind idealerweise gleichverteilt, da programmiertechnisch allerdings mit Pseudozufallszahlengeneratoren gearbeitet wird und die Amplitude durch die maximale Darstellungsgenauigkeit in Rechnern begrenzt ist, handelt es sich in diesem Fall um rosa Rauschen. Die neuen Individuen (Kinder) werden zusätzlich mit weiteren Symmetrie erzeugenden Regeln modifiziert. Welche Regel jeweils auf ein Individuum angewendet wird, wird ebenfalls durch eine Zufallsauswahl bestimmt, die jeder Regel eine Auswahl mit gleicher Wahrscheinlichkeit der Auswahl zuordnet. Diese Regeln zielen darauf ab, nicht nur die CNN Parameter der Templates zu optimieren, sondern auch die bei vielen CNN-Anwendungen inhärente Symmetrie bezüglich der Template-Matrizen, unterstützend im Optimierungsprozess zu berücksichtigen. Die symmetrieezeugenden Operationen sind für eine  $3 \times 3$  Matrix unter anderem Spiegelungen an der mittleren Spalte (vergleiche Bild 5.2) oder mittleren Zeile, bzw. eine punktsymmetrische Spiegelung (aus Gründen der Übersichtlichkeit hier nicht gezeigt) am mittleren Element. Weitere implementierte Symmetrieeoperationen sind die Spiegelung an einer der Diagonalen bzw. Teilspeiegelungen der Template-Elemente und die spezielle Regel der Anwendung keiner Symmetrie. Durch die Anwendung der Regeln kommt es dann über die Generationen der Optimierung zu einer Bevorzugung von Parametersätzen mit Symmetrieeigenschaften, die dem zu



**Bild 5.2:** Beispiel einer Symmetrieoperation: Spiegelung an der Mittelachse.

behandelnden Optimierungsproblem am ehesten gerecht wird. Hierbei ist wichtig, daß man dem Verfahren a priori kein Vorwissen über die mögliche Symmetrie des Optimierungsproblems vorgeben muß, sofern eine Symmetrie in der zu optimierenden Problemstellung vorhanden ist, wird diese durch SSAPE bei der Optimierung mit berücksichtigt. Ein Vorwissen über Symmetrien der Problemstellung kann allerdings durch Einschränkung der möglichen Auswahl an anzuwendenden Symmetrieregeln mit berücksichtigt werden. Durch Selektion der  $k$  Individuen mit den jeweils kleinsten Fehlern innerhalb einer Generation als Eltern der nächsten Generation, bildet sich über die Iterationen des Optimierungsverfahrens eine Population aus, in welcher eine Symmetrie bevorzugt auftritt, sofern eine solche in der Lösung der Problemstellung enthalten ist. Diese Symmetrie ist also nicht fest vorgegeben, sondern wird durch das Optimierungsverfahren erzeugt, was insbesondere dann von Vorteil ist, wenn Bauteiletoleranzen eine feste Vorgabe von Symmetrien behindern bzw. eine bestimmte symmetrische Template wegen Bauteiletoleranzen keine optimale Lösung sein kann.

Ein weiteres evolutionäres Optimierungsverfahren, Differential Evolution (DE), welches von R. Storn und K. Price [41][42] eingeführt wurde, dient in den folgenden Untersuchungen als Vergleichsverfahren. Dazu wurde auch der DE-Algorithmus auf dem EyeRIS-System implementiert. Das DE-Verfahren bildet jedes Individuum der fol-



genden Generation aus mehreren zufällig gewählten Individuen der momentanen Generation. In einer einfachen Implementierung von DE wird zur Erzeugung eines neuen Parametervektors der folgenden Generation zuerst aus zwei Parametervektoren der momentanen Generation eine gewichtete Differenz gebildet. Diese wird anschließend zu einem weiteren Parametervektor der momentanen Generation unter Berücksichtigung eines Crossover-Operators [29] addiert. Weiterhin wurden Simulationen mit SSAPE, DE sowie mit den Optimierungsverfahren Simulated Annealing und Broyden-Fletcher-Goldfarb-Shannon (BFGS) [31] durchgeführt, um die Ausführungsgeschwindigkeiten der Verfahren miteinander zu vergleichen. In Tabelle 5.1 sind die Ergebnisse tabellarisch zusammengefaßt. Das BFGS Verfahren war hierbei bei den Simulationen den anderen Verfahren in der Ausführungsgeschwindigkeit überlegen, ließ sich aber auf dem EyeRIS System als auch in Simulationen mit 8 bit nicht untersuchen, da es ein Gradienten basiertes Optimierungsverfahren ist und ein Gradient sich mit der reduzierten diskreten Wertedarstellung nicht immer berechnen ließ und zum Abbruch des Verfahrens führte.

Während der Berechnungen auf dem EyeRIS System mußten die Berechnungen des Öfteren neu gestartet werden, da das System ab und an einen Reset benötigte. Ein möglicher Grund für die genannten Aussetzer des Systems könnten thermische Probleme sein, die zur leichten Überhitzung führten. Dies führte zu real längeren Laufzeiten als die in Tabelle 5.1 angegebenen, reinen Rechenzeiten. Zum einen beliefen sich dadurch die Rechenzeiten auf drei bis sechs Wochen für einzelne Kanäle, zum anderen bestand die Notwendigkeit Segmente der Datensätze wiederholt berechnen zu müssen.

Verfahren	Netzwerkberechnungen/Min
DE Chip	5.79
SSAPE Chip	6.56
DE	16.35
DE 8 bit	57.04
SSAPE	4.02
SSAPE 8 bit	15,97
BFGS	236.87
Simulated Annealing	16.8

**Tabelle 5.1:** Mittlere Anzahl von Neuberechnungen von Gewichtswerten pro Minute. Das BFGS Verfahren ist das schnellste von allen, benötigt aber die Berechnung eines Gradienten, der bei Diskretisierung auf 8 bit nicht immer korrekt berechnet werden kann und zu frühzeitigem Abbruch des Verfahrens führt.

### 5.3 Der SSAPE-Algorithmus auf dem EyeRIS-System

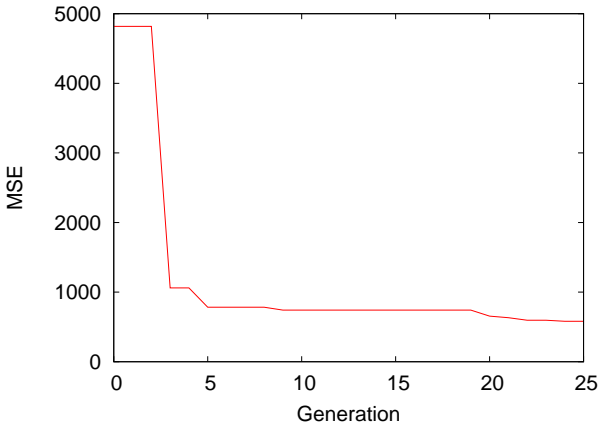
In diesem Abschnitt werden exemplarisch Ergebnisse gezeigt, die aus der Anwendung der vorgestellten Optimierungsverfahren auf einfache Problemstellungen der Bildverarbeitung mit CNN resultieren. Dies betrifft die sogenannte Hole-Filler Template, die Kantendetektion und die begrenzte Diffusion. Die Hole-Filler Operation stellt eine propagierende Operation dar, wobei Objekte eines binären Bildes mit schwarzen Pixeln ausgefüllt werden, so daß die Rechenzeiten bis zum Erreichen eines stabilen Endzustandes abhängig von der Fläche des zu füllenden Objektes im Bild sind.

Bei der binären Kantendetektion werden sämtliche Pixel, die nicht die Kante eines Bildobjekts darstellen, zu weißen Pixeln im stabilen Endzustand des Netzwerks, die Kantenpixel bleiben schwarz. Im

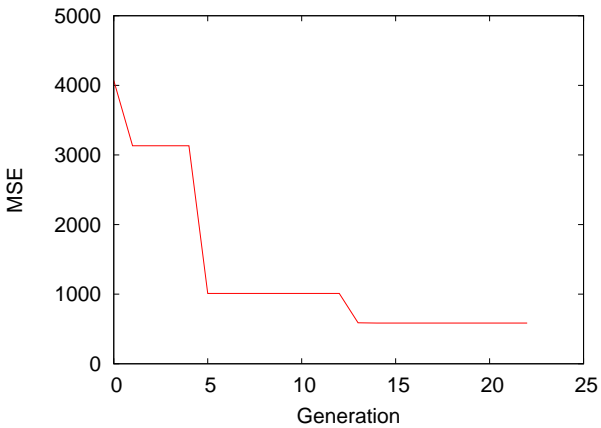
folgenden werden die Optimierungsverfahren SSAPE und DE auf ihre Fähigkeit zur Optimierung auf dem EyeRIS-System untersucht. Hierbei setzen sich die Parametersätze nicht nur aus Templateparametern zusammen (im Falle einer  $3 \times 3$  Tempalte also 9 Parametern), sondern es können auch schaltungsbedingte Größen in die Optimierung mit einbezogen werden, wie im Falle des EyeRIS Systems eine Gewichtung des Biaswertes  $V_{in}$ , der in CNN so nicht zu finden ist. Auch wenn vom Hersteller des EyeRIS-Systems davon abgesehen wird, können auch weitere Größen wie maximale und minimale Werte der Spannungsreferenzen für Gewichte und Zustandswerte mit optimiert werden. Während der Optimierung wird jeder Parametersatz einer Generation auf dem EyeRIS-System unter Verwendung unterschiedlicher Bilder, welche die sogenannte Trainingsmenge bilden, zur Berechnung herangezogen; dabei wird der jeweilige MSE ermittelt. Anschließend werden die optimierten Parameter zur Überprüfung auf weitere Bilder angewandt, welche nicht in der Trainingsmenge enthalten sind. In Bild 5.3 ist die zeitliche Entwicklung des MSE gegen den Iterationsschritt (Generation) für die Hole-Filler-Verarbeitung gezeigt. Wie erkennbar ist, nimmt der MSE bei Anwendung des DE-Algorithmus anfangs sehr rasch ab und verweilt dann lange auf einem Fehlerniveau. Erst nach der 18ten Generation ist eine weitere Absenkung des MSE zu beobachten. Ein ähnliches Verhalten zeigt auch der SSAPE-Algorithmus. In Bild 5.4 ist ein Ergebnis gezeigt, welches nach Anwendung eines optimierten Parametersatzes

$$\mathbf{W}x_1 = \begin{pmatrix} 87 & 149 & 144 \\ 37 & 9 & 116 \\ 88 & 87 & 80 \end{pmatrix}, \quad V_{in} = 155 \quad (5.6)$$

auf ein nicht in der Trainingsmenge enthaltenes Bild resultiert. Hier ist auch ein leichter Grauschleier im Ergebnisbild zu erkennen, der auf schaltungstechnische Einflußfaktoren der Realisierung zurückzuführen ist. Bild 5.5 zeigt einen Vergleich zur Kantendetektion für die Konvergenz des SSAPE-Algorithmus bei 9 bzw. 13 zu optimie-

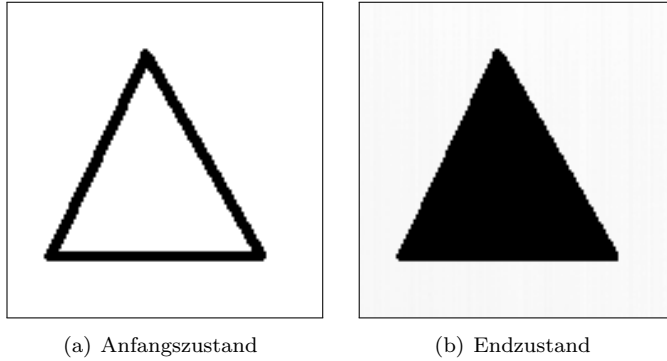


(a) DE



(b) SSAPE

**Bild 5.3:** Zeitliche Entwicklung des MSE als Funktion der Generationen (Iterationsschritte) für (a) DE und (b) SSAPE-Algorithmus (11 Parameter) für Hole-Filler-Verarbeitung.



**Bild 5.4:** Beispiel für die Hole-Filler-Verarbeitung; (a) Anfangszustand; (b) Resultat nach Anwendung der Optimierung auf dem EyeRIS-System.

render Parameter, wobei die Template

$$\mathbf{W}x_1 = \begin{pmatrix} 222 & 134 & 229 \\ 49 & 36 & 50 \\ 223 & 161 & 217 \end{pmatrix} \quad (5.7)$$

für 9 Parameter und

$$\mathbf{W}x_1 = \begin{pmatrix} 179 & 147 & 186 \\ 141 & 8 & 138 \\ 167 & 155 & 165 \end{pmatrix}, \quad (5.8)$$

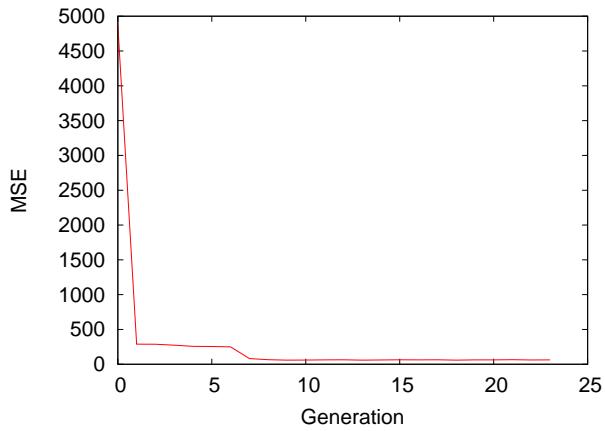
$$V_{in} = 175, \quad Wx_2 = 4, \quad Wx_3 = 184, \quad Wx_4 = 85$$

für 13 Parameter durch die Optimierung erhalten wurden. In den bisher betrachteten Fällen zeigt sich der Trend, mehr Iterationsschritte zu benötigen, um einen vergleichbar guten Wert des MSE zu erreichen, wenn mehr Parameter zu optimieren sind. Im Fall von 9 Parametern kann ein Abfall des MSE bereits nach der 5. Generation beobachtet werden, während im Fall von 13 zu optimierenden

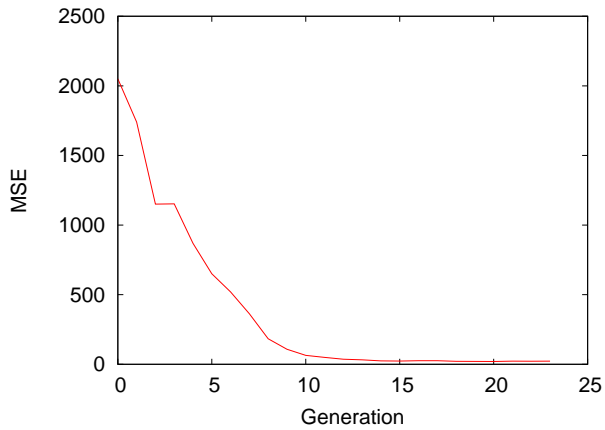
Parametern schon 10 Generationen nötig werden, um einen ähnlich deutlichen Abfall des MSE zu erreichen. Die begrenzte Diffusion ist eine komplexe Problemstellung, da hierbei sowohl Grauwertbilder als auch Binärbilder verarbeitet werden. Hier wird als Eingabe ein Grauwertbild und als Anfangszustand ein binärwertiges Bild, welches kleine schwarze Punkte aus mehreren schwarzen Pixeln als Markierer enthält verwendet, vergleiche Bild 5.6. Ein Markierer an einer Stelle, welche im Eingabebild einen Grauwert enthält, der über einem durch den Bias  $z$  bestimmten Schwellwert liegt, breitet sich von dieser Stelle ausgehend aus. An Stellen, an denen der Grauwert des Eingangsbildes kleiner als der Schwellwert ist, hört die Ausbreitung des Markierers auf. Markierer an Stellen, an denen von Anfang an der Schwellwert nicht genügend groß war, verschwinden während der analogen Operation. Die Ergebnisse der Optimierung der begrenzten Diffusion sind in Bild 5.6 in zwei Spalten dargestellt. In beiden Fällen wurde das gleiche Eingangsbild benutzt, während für den Anfangszustand zwei unterschiedliche binäre Bilder verwendet wurden. Während in der rechten Spalte der Anfangszustand aus nur einem Markierer besteht, wurden für den in der linken Spalte gezeigten Fall drei Markierer verwendet. Zum Vergleich sind Ergebnisse eines Trainings für die 3. Generation, welche schon zu einem guten Resultat führt, und der 23. Generation gezeigt, welche nahe an das gewünschte Resultat heranreicht. Die Entwicklung des MSE über die Generationen/Iterationen des jeweils besten Parametersatzes einer Generation ist in Bild 5.7 gezeigt.

## 5.4 SSAPE mit Graustufenbildern

Die Ergebnisse des vorhergehenden Abschnitts wurden unter der Prämisse gewonnen, daß die Ergebnisbilder schwarz/weiß sein sollten. Im Folgenden wurde das SSAPE auch auf Problemstellungen



(a) 9 Parameter

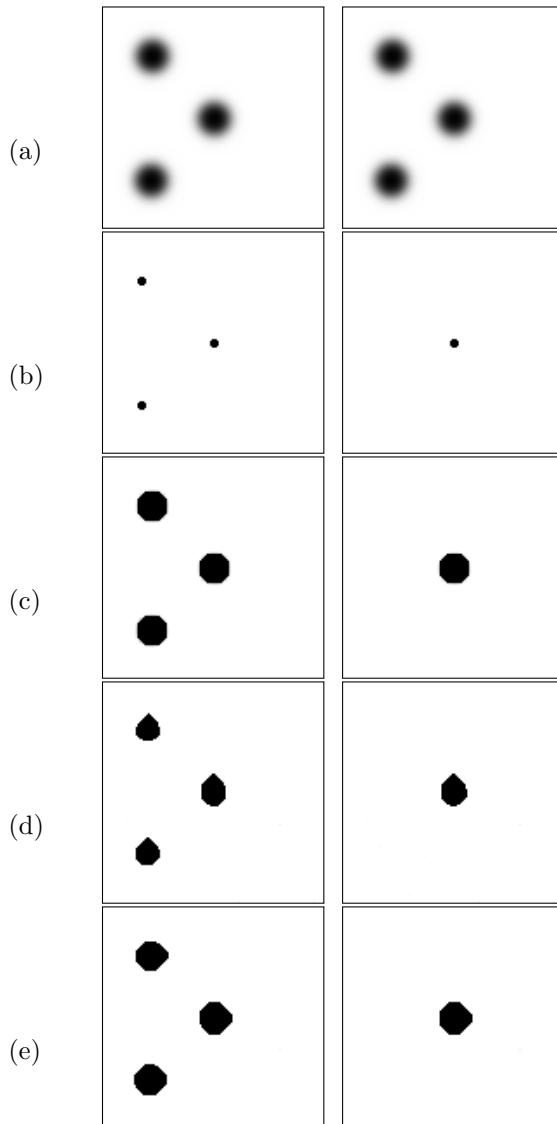


(b) 13 Parameter

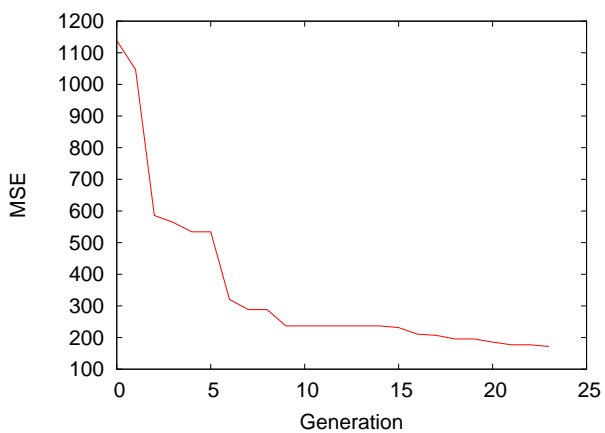
**Bild 5.5:** Kantendetektion: (a) 9 Parameter; (b) 13 Parameter mit SSAPE durchgeführt.

angewandt, in denen der Wertebereich sowohl der Eingangsdaten als auch der Ausgangsdaten durch ein kontinuierliches Grauwertintervall repräsentiert wurde. Hierbei kann aufgrund der erreichbaren Rechengenauigkeit und der Darstellungsgenauigkeit nur näherungsweise von kontinuierlichen Grauwertverläufen ausgegangen werden. Während in der Simulation mit Fließkommazahlen doppelter Genauigkeit gerechnet wird und der Darstellungsbereich voll ausgenutzt wird, ist die Stufung der Grauwerte bei den zur Zeit verfügbaren CNN-Hardwarerealisierungen auf die 8 bit Genauigkeit der AD/DA-Wandler beschränkt, was 256 Grauwertstufen entspricht. Die Genauigkeit der in den Untersuchungen verwendeten Hardware wurde für Grauwertbilder auch schon in Abschnitt 3.2 untersucht und wirken sich besonders auf die Ergebnisse aus, die eine hohe Genauigkeit fordern. Die Vorhersage epileptischer Anfälle zählt beispielsweise dazu und die hierzu gehörenden Ergebnisse werden in Abschnitt 7.1 diskutiert.





**Bild 5.6:** Von oben nach unten: (a) Bild in der Eingabe des CNN mit Markierern, (b) Anfangszustand des CNN, (c) Referenzbild, (d) Ergebnis des besten Individuums in der 3. Generation, (e) Ergebnis des besten Individuums in der 23. Generation.



**Bild 5.7:** MSE in Abhängigkeit von der Anzahl der Iterationen für die Optimierung zur begrenzten Diffusion mit SSAPE.

# 6 Statistische Untersuchungsmethoden

In diesem Kapitel werden zwei statistische Auswertungsmethoden vorgestellt, welche unter anderem für die qualitative Begutachtung und quantitative Bewertung von Meßdaten herangezogen werden können. Das erste dieser Verfahren ist die sogenannte Receiver-Operating-Characteristic [43, 44], kurz ROC genannt, welche entwickelt wurde, um die Leistungsfähigkeit von Klassifizierern zu bewerten. Ursprünglich für die Untersuchung von Fehlerraten in der Radartechnik entwickelt, stellte sich die ROC-Analyse auch in anderen Untersuchungsgebieten z.B. in der Medizinforschung als nützlich heraus [45] und fand schließlich sogar ihren Weg in Arbeiten zur künstlichen Intelligenz [46]. Das zweite hier vorgestellte Verfahren dient der Erzeugung von künstlichen Datensätzen, sogenannten Surrogaten und wird in Kombination mit der ROC-Analyse in Hypothesentests angewendet.

## 6.1 Die Receiver Operating Characteristic (ROC)

Die ROC-Analyse wird angewendet, wenn eine Aussage über die Güte eines Testverfahrens bzw. eines Bewertungsverfahrens gemacht werden soll, und ist somit ein Klassifizierungsverfahren. Zumeist wird das Vorhandensein oder Nichtvorhandensein eines Merkmals mittels eines Schwellwertes betrachtet. Angenommen, man hat einen

hier nicht näher spezifizierten Prozess oder Daten eines solchen, für den das Vorhandensein einer Eigenschaft, ein Merkmal, mittels eines Schwellwertes detektiert werden kann. Der Einfachheit halber sei vorausgesetzt, daß hierfür nur ein Schwellwert unter- bzw. überschritten werden muß. Mit Hilfe der ROC-Analyse kann untersucht werden, wie gut das Verfahren das Vorhandensein des Merkmals vom Nichtvorhandensein mittels eines Schwellwertes trennen (diskriminieren) kann. Bei jeder Einzeldetektion gibt es nun vier Möglichkeiten, die zu berücksichtigen sind:

- Werte, die oberhalb des Schwellwertes liegen und tatsächlich das Vorhandensein des Merkmals anzeigen. Diese sind die sogenannten True Positives ( $TP$ ), also korrekt detektierte.
- Werte, die unterhalb des Schwellwertes liegen, wobei das Merkmal tatsächlich nicht vorhanden ist. Diese sind die sogenannten True Negatives ( $TN$ ).
- Werte, die oberhalb des Schwellwertes liegen, bei denen aber das Merkmal nicht vorliegt. Diese sind die sogenannten False Positives ( $FP$ ), da sie fälschlicherweise das Vorhandensein des Merkmals detektiert haben, obwohl es nicht vorliegt, auch Fehler 1. Art genannt.
- Werte, die unterhalb des Schwellwertes liegen, obwohl das Merkmal vorliegt. Diese werden als False Negative ( $FN$ ) bezeichnet, da sie auf das Nichtvorhandensein des Merkmals hindeuten, es aber tatsächlich vorliegt, was auch als Fehler 2. Art bezeichnet wird.

Im Idealfall kann das Untersuchungsverfahren das Vorhandensein des Merkmals sauber vom Nichtvorhandensein des Merkmals trennen und es treten keine Fehldetektionen  $FP$  und  $FN$  auf. Unter realen Bedingungen ist dies meist aufgrund von Rauschen und anderen Schwankungen jedoch nicht möglich. Wird das Auftreten der  $TP$ ,  $FP$ ,  $TN$  und  $FN$  normiert, so ergeben sich die entsprechenden Ra-

ten  $tp$ ,  $fp$ ,  $tn$  und  $fn$  zu:

$$tp = \frac{TP}{TP + FN} \quad (6.1)$$

auch  $tp$ -Rate = Sensitivität genannt;

$$tn = \frac{TN}{FP + TN} \quad (6.2)$$

auch  $tn$ -Rate = Spezifität genannt, sowie

$$fp = \frac{FP}{FP + TN} \quad (6.3)$$

und

$$fn = \frac{FN}{TP + FN}. \quad (6.4)$$

Hier gilt durch die Normierung:

$$tp + fn = 1 \quad \text{und} \quad fp + tn = 1 \quad (6.5)$$

wodurch sich die Spezifität auch als

$$tn = 1 - fp \quad (6.6)$$

schreiben läßt.

Weiterhin ergibt sich aus  $P = TP + FN$  und  $N = FP + TN$ :

$$tp + fp = \frac{TP}{P} + \frac{FP}{N} \quad (6.7)$$

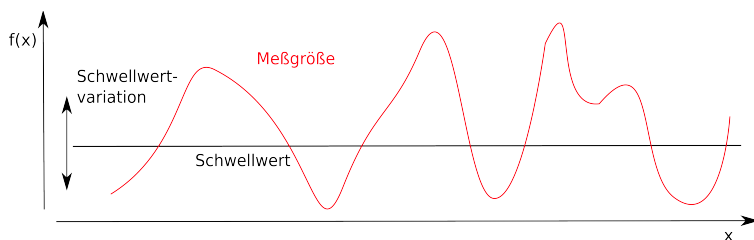
und

$$tn + fn = \frac{TN}{N} + \frac{FN}{P}, \quad (6.8)$$

welche sich im Idealfall, daß keine Fehldetektionen erfolgen, jeweils zu 1 summieren. In Tabelle 6.1 sind die genannten möglichen Kombinationen noch einmal zusammengestellt. Es ist nun zu klären, bei welchem Schwellwert das Verfahren zur Detektion des Merkmals die wenigsten Fehlklassifikationen macht, also welcher Schwellwert der

	Merkmal vorhanden	Merkmal nicht vorhanden
Merkmal detektiert	tp	fp
Merkmal nicht detektiert	fn	tn
Summe	1	1

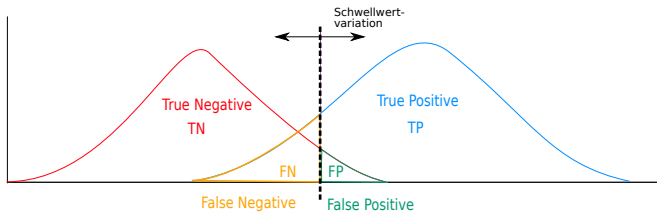
**Tabelle 6.1:** Tabelle der vier Kombinationsmöglichkeiten für die ROC.



**Bild 6.1:** Signal mit Schwellwert.

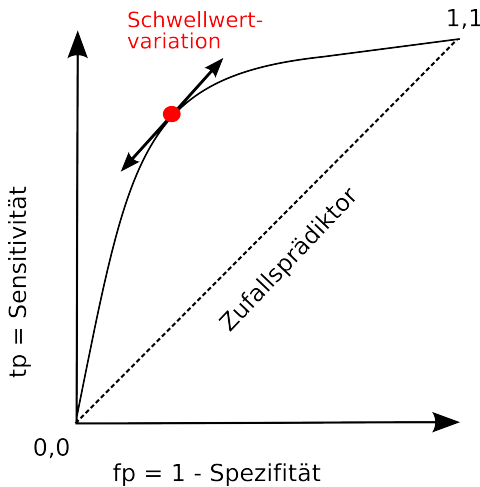
optimale Schwellwert ist. Hier sollten die Werte für False Positive und False Negative die jeweils kleinsten Werte annehmen, im Optimum sogar Null sein. Um das Optimum zu finden, wird der Schwellwert, wie in Bild 6.1 gezeigt, variiert.

Für jeden Schwellwert ergibt sich dann eine Tabelle analog zu Tabelle 6.1; hieraus können dann zwecks besserer Veranschaulichung von True Positive und True Negative zugeordnete Verteilungsdichte-Graphen erzeugt werden. Für einen Schwellwert, der das Auftreten des Merkmals eindeutig detektieren kann, überlappen sich diese Verteilungsdichten nicht. In den meisten Fällen wird man aber von Null verschiedene True Negative- und False Positive Werte erhalten, die aus Überschneidungen der Verteilungsdichtegraphen resultieren (siehe Bild 6.2). Um eine einzelne Maßzahl zu erhalten, welche die



**Bild 6.2:** Verteilungen der  $TP$ ,  $TN$ , und die Überschneidung aus der sich zusammen mit dem Schwellwert  $FP$  und  $FN$  ergeben.

Eignung des Verfahrens bewertet, wird die True Positive Rate gegen die False Positive Rate aufgetragen (siehe Bild 6.3). Da die Raten normiert sind, ergibt sich dabei ein maximaler Flächeninhalt von 1. Die Fläche unter der ROC-Kurve, auch „Area Under the Curve“, kurz AUC-Wert genannt, wird oft verwendet, um die Güte der ROC-Analyse zu spezifizieren. Ein zufälliger Prädiktor würde durch eine Gerade von  $(0, 0)$  bis  $(1, 1)$  im ROC-Graphen, also eine Winkelhalbierende, dargestellt werden und die Fläche genau halbieren, was einem AUC-Wert von 0.5 entspricht; dies ist in Bild 6.3 durch die Diagonale dargestellt. Aus der Normierung der Fläche ergibt sich auch, daß der AUC-Wert nur Werte im Intervall  $[0; 1]$  annehmen kann. Je weiter ein Prädiktor vom Zufallsprädiktor abweicht, desto besser ist er geeignet, die untersuchte Größe zu erkennen. Liegen alle Punkte einer ROC-Analyse unterhalb der des Zufallsprädiktors, dann kann man das Detektionskriterium „negieren“ und so eine ROC-Kurve oberhalb der Diagonalen des Zufallsprädiktors erhalten und somit einen Wert für die AUC größer als 0.5. Der Betrag  $|0.5 - AUC|$  ist also ein Maß für die Güte der Erkennung des untersuchten Merkmals.



**Bild 6.3:** Area Under the Curve (AUC). Durch die Variation des Schwellwertes wird der die AUC-Fläche bestimmende Graph erzeugt. Jeder Punkt auf dem Graphen entspricht somit den True Positive und False Positive Werten für einen Schwellwert.

## 6.2 Nullhypothesentest und Surrogate

Ein Hypothesentest, welcher verwendet werden kann, um die Richtigkeit einer Annahme zu prüfen, basiert auf der Aufstellung einer sogenannten Nullhypothese  $H_0$ , welche die eigentliche Annahme negiert. Das heißt, es wird unterstellt, daß eine Annahme mittels ROC-Analyse und einer diskriminierenden Größe, z.B. einem Schwellwert, nicht gilt. Bestätigt sich dann das Gegenteil, wird die Nullhypothese verworfen bzw. zurückgewiesen, und die ursprüngliche Annahme, die Alternativhypothese  $H_1$ , ist zumindest wahrscheinlicher geworden. Beide Hypothesen müssen daher disjunkte Aussagen treffen, welche sich klar unterscheiden. Als Beispiel soll ein Prozess betrachtet werden, von dem angenommen wird, daß er nichtlinear sei. Die



Nullhypothese lautet dann, daß der Prozess nicht nichtlinear, also linear ist.

Im Folgenden wird nun noch der Begriff der „Surrogate“ benötigt. Das Wort Surrogat kommt aus dem Lateinischen und bedeutet soviel wie Ersatz oder im – hier betrachteten Fall – Austauschdaten oder Ersatzdaten. Die Erzeugung dieser Surrogate muß immer unter Berücksichtigung der zuvor aufgestellten Nullhypothese erfolgen, d.h. im obigen Beispiel müssen die Surrogate die durch die Nullhypothese vorgegebene lineare Bedingung erfüllen, nicht aber die nichtlinearen. Einige Verfahren für die Erzeugung von Surrogaten [47, 48] sind:

- Bootstrap Verfahren, bei dem die Statistik des Prozesses analytisch erlangt werden kann, um Surrogate zu erzeugen,
- Amplitude Adjusted Fourier Transform Algorithm,
- Windowed Fourier Transform Algorithm,
- Seizure Time Surrogate,
- Measure Profile Surrogate

Wird nun ein Verfahren zur Detektion von nichtlinearem Systemverhalten auf das Originalsignal und auf die Surrogate angewandt, so sollte das Verfahren, wenn das untersuchte System nichtlinear sein sollte, eine bessere Diskriminierung ermöglichen, als dies bei den Surrogaten der Fall ist. Wie bei der ROC Analyse erläutert, kann es dabei zu Fehleinordnungen kommen, den False Positives und den False Negatives.

Ein Fehler 1. Art liegt hierbei dann vor, wenn man die Alternativhypothese  $H_1$  annimmt und die Nullhypothese  $H_0$  zurückweist, obwohl die Nullhypothese gilt, also False Positive. Ein Fehler 2. Art liegt vor, wenn die Nullhypothese beibehalten wird, obwohl faktisch die Alternativhypothese gilt, also False Negative. Beide führen zu Fehlklassifizierungen.

Um eine Aussage über die Güte eines Verfahrens und seiner Fehlklassifizierungen zu erhalten, wird nun ein Signifikanzlevel  $\alpha$  als zu erfüllende Bedingung festgelegt, der die Signifikanz einer Detektion mit dem untersuchten Verfahren spezifiziert. Um einen Signifikanzlevel von  $\alpha = 5\%$  zu erhalten, ist es nötig,  $M = 20$  Datensätze zu untersuchen. Dies ergibt sich aus der Überlegung, daß

$$\alpha = 5\% = \frac{1}{20}$$

und somit

$$M = \frac{1}{\alpha} = 20 \quad (6.9)$$

ist. Da die ursprünglichen Daten hierbei in den 20 Datensätzen enthalten sind, muß man nur 19 zusätzliche Surrogat-Datensätze erzeugen [49, 50]. Dies bedeutet, daß mindestens 20 Datensätze berücksichtigt werden müssen um mit 5% oder weniger die Nullhypothese zu bestätigen bzw. daß man mit einer Wahrscheinlichkeit von 95% sicher sein kann, daß die Nullhypothese zu Recht zurückgewiesen wurde.

Nicht immer ist eine vollständige statistische Auswertung möglich, weshalb man dann auf eine Stichprobenauswertung angewiesen ist. Wenn ein Element der Stichprobe nur einen von zwei Zuständen, bzw. Elementarereignissen, annehmen kann, wie z.B. Probe erfüllt Kriterium und Probe erfüllt Kriterium nicht, handelt es sich um ein Bernoulliexperiment. Die Auftrittswahrscheinlichkeit ist hierbei durch die Binomialverteilung [51, 52, 53]

$$P(\nu, n) = b_{n,p}(\nu) = \binom{n}{\nu} p^\nu q^{n-\nu}, \quad (6.10)$$

mit  $n$  Gesamtanzahl aller Versuche,  $\nu$  Anzahl erfolgreicher Versuche,  $p$  der Auftrittshäufigkeit eines Erfolges und  $q = (1-p)$  die eines Mißerfolges, gegeben. Hiermit läßt sich die kumulierte Bernoulliverteilung [54] als die Summe über  $k$  Wahrscheinlichkeiten mit

$$P(\nu \leq k, n) \leq \sum_{i=0}^k \binom{n}{i} p^i q^{n-i} \quad (6.11)$$

für das Auftreten von einem Erfolg, zwei Erfolgen, 3 Erfolgen etc. beschreiben. Bei einer Stichprobenauswahl kann hiermit berechnet werden wie groß die Wahrscheinlichkeit ist, ein entsprechendes Ergebnis zu erhalten. Für die Abschätzung des Fehlers 1. Art bei einem Hypotesentest ergibt sich hieraus die Bedingung

$$1 - \sum_{i=0}^{k_g} \binom{n}{i} p_0^i q_0^{n-i} \leq \alpha, \quad (6.12)$$

wobei  $k_g$  den Grenzwert für  $k$  angibt bei dem die Bedingung gerade noch erfüllt ist. Ein Annehmen der Nullhypothese  $H_0$ , basierend auf diesem Kriterium, obwohl die Alternativhypothese  $H_1$  gilt, ist ein Fehler 2. Art, auch als  $\beta$ -Fehler bezeichnet. Wird ein  $\beta$  vorgegeben, so kann, unter beibehalten von  $k_g$ , die Bedingung

$$P(\nu \leq k_g, n | H_1) = \sum_{i=0}^{k_g} \binom{n}{i} p_1^i q_1^{n-i} \leq \beta, \quad (6.13)$$

unter der Voraussetzung, daß  $H_1$  zutrifft, als Berechnungsgrundlage für die Abschätzung für das maximale  $p_1$  verwendet werden, so daß die in der Stichprobe  $s$  ermittelte Auftrittshäufigkeit von Erfolgen  $p_s > p_1$  mit einer Wahrscheinlichkeit  $\leq \beta$  gerade noch gilt [54]. Beide Abschätzungen können effektiv mit heute verfügbaren Tabellenkalkulationsprogrammen [55] durchgeführt werden. Die Funktion *critbinom*( $n, p, \alpha$ ) mit  $n$  Anzahl der Versuche,  $p$  Wahrscheinlichkeit für einen Erfolg und  $\alpha$  als Kriterium für die Signifikanz, liefert die Anzahl der Versuche für welche der Wert der Binomialverteilung gerade noch größer oder gleich  $\alpha$  ist [56]. Hiermit läßt sich für die Detektionsrate ein Grenzwert bestimmen und gegen die Nullhypothese prüfen. Die Funktion *binomdist*( $s, n, p, cumulativ$ ) liefert mit  $s$  Anzahl der Erfolge,  $n$  Gesamtanzahl der Versuche,  $p$  Wahrscheinlichkeit für eine Erfolg und mit *cumulativ* = 0 als Auswahlkriterium, die Wahrscheinlichkeit für das exakte Auftreten von  $s$  Erfolgen sowie für *cumulativ* = 1 die kumulierte Wahrscheinlichkeit [57] und kann für die Abschätzung für  $p_1$  verwendet werden sodaß Formel 6.13 erfüllt ist.



## 7 Anwendungsbereiche

In diesem Kapitel werden zwei mögliche Anwendungsgebiete für CNN vorgestellt. Hierbei wird besonderes Augenmerk auf die Umsetzung der Algorithmen mit schaltungstechnischen Realisierungen von CNN gelegt.

### 7.1 Vorhersage epileptischer Anfälle

Epilepsie als Erkrankung wird schon in babylonischen Texten aus der Mitte des zweiten Jahrtausends vor Christus erwähnt [58] und ist somit eine der ältesten dokumentierten Erkrankungen des zentralen Nervensystems. Heute ist sie mit eine der häufigsten chronischen Erkrankungen des zentralen Nervensystems. Epilepsie zeichnet sich durch eine Anomalie in der Synchronisation der neuronalen Aktivität aus [59], die von Krämpfen, Bewußtseinsverlust oder Gleichgewichtsstörungen begleitet werden kann. Die Erscheinungsformen der epileptischen Anfälle werden allgemein in unterschiedliche Arten klassifiziert, die je nach Schwere der Erscheinung spezifisch behandelt werden müssen. Eine Art der Epilepsie ist die fokale Epilepsie<sup>1</sup>, die ihren Namen daher hat, daß ein Anfall von einem Anfallsherd im Gehirn, dem sogenannten epileptogenen Areal – Fokus – ausgeht, von dem aus sich der Anfallszustand ausbreitet. Die Behandlung wird heutzutage mittels Medikamenten (Antikonvulsiva)

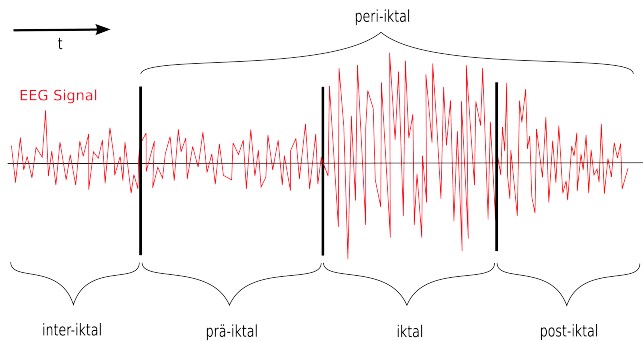
---

<sup>1</sup>EEG-Signale anderer Epilepsieformen standen für die hier durchgeführten Untersuchungen nicht zur Verfügung.

durchgeführt, die allerdings nicht in allen Fällen angewendet werden können. Unverträglichkeiten oder die Gefahr einer Abhängigkeit [60] bedingen die Erforschung neuer Behandlungsmethoden. Eine alternative Behandlungsmethode ist die mögliche gezielte Beeinflussung von Hirnarealen mit elektrischer Stimulation oder gezielte lokale Medikamentierung durch Mikropumpen in einem kleinen Hirnareal. Um solche Maßnahmen einsetzen zu können, ist es notwendig, das Nehen eines epileptischen Anfalls zu detektieren. Die Prädiktion von epileptischen Anfällen durch Algorithmen, welche die als vorhanden angenommenen Vorboten eines epileptischen Anfalls in Hirnstromverläufen – einem Elektroencephalogramm, kurz EEG – detektieren können, ist eine Zielsetzung, die bereits mit verschiedenen Algorithmen angegangen wurde. Die Zustände, die sich z.B. in einem solchen EEG-Signal finden lassen, gliedern sich nach [61] in

- den inter-iktalen-Zustand (von lat. ictus: betroffen, beunruhigt), anfallsfreier Zustand zwischen Anfällen,
- den prä-iktalen-Zustand, Voranfallszustand,
- den peri-iktalen-Zustand, Zustand um einen Anfall herum einschließlich des Anfalls,
- den iktalen-Zustand, Anfallszustand und
- den post-iktalen-Zustand, Zeit nach einem Anfall, in der das EEG sich noch nicht normalisiert hat.

Hierbei wird davon ausgegangen, daß der Übergang vom anfallsfreien – inter-iktalen – Zustand in den Anfall – iktaler Zustand – ein gradueller Prozess ist [62, 63], in welchem Vorboten in einem Zeitbereich vor dem Anfall – prä-iktaler Bereich – detektiert werden können. Verschiedene Untersuchungen haben gezeigt, daß ein Voranfallszustand bei Patienten mit fokaler Epilepsie vorhanden ist und mittels multivariater Maße erkannt werden kann [64, 65, 66, 67, 68, 69, 70]. Nach dem iktalen Bereich im Signal schließt sich noch ein Signalbereich an, der als post-iktal bezeichnet wird, bevor man das Signal wieder als inter-iktal bezeichnen kann. Hierbei kann sich der prä-iktale



**Bild 7.1:** Für eine Prädiktion eines epileptischen Anfalls wichtige Zeitbereiche in einem EEG-Signal.

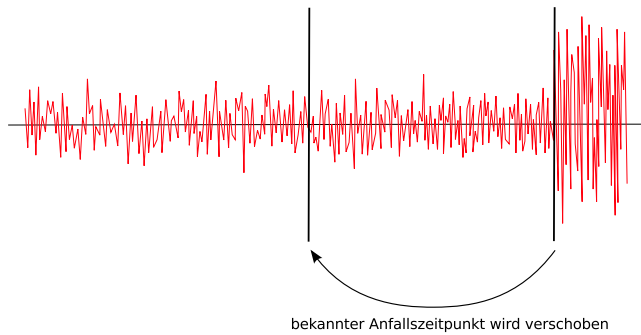
und post-iktale Bereich auf 24 Stunden und mehr ausdehnen [71]. In dieser Arbeit werden der prä-iktale, ictale und post-iktale Bereich unter dem Begriff peri-iktal zusammengefaßt.

Die Anwendung von CNN in der Medizintechnik zur Vorhersage epileptischer Anfälle ist ein Gebiet, in dem sich ihre Verarbeitungsparallelität wie auch die kleine, energiesparende analoge Bauweise als besonderer Vorteil anbietet. In diesem Kapitel werden Implementierungen zweier Algorithmen zur Prädiktion von Epilepsie auf dem EyeRIS-System vorgestellt. Die Algorithmen wurden in [72, 73, 74, 75] eingeführt und auf ihre Tauglichkeit für die Prädiktion von Epilepsie untersucht, nachfolgend wird nur auf die Umsetzung auf dem EyeRIS-System eingegangen. Die Implementierung auf einem solchen System soll hierbei die Machbarkeit eines kleinen tragbaren Systems für die Vorhersage epileptischer Anfälle demonstrieren. Die Güte der Ergebnisse wird mit Surrogaten überprüft; hierfür können die sogenannten Seizure Time Surrogate [48, 76] bzw. die Measure Profile Surrogate [47, 76] verwendet werden, die sich durch folgende Vorgehensweisen bilden lassen:

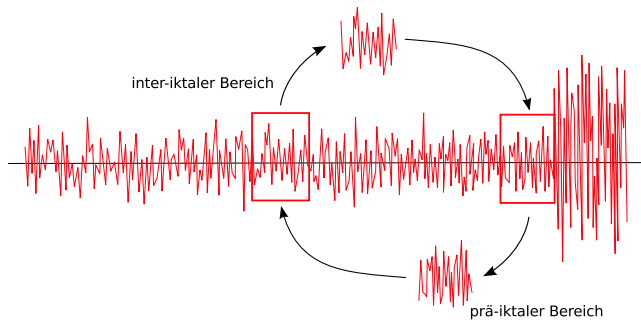
- Um Seizure Time Surrogate zu erzeugen, werden die bekannten Anfallszeitpunkte verschoben, was in Bild 7.2 gezeigt ist. Man nimmt also einen anderen Anfallszeitpunkt als den tatsächlichen Anfallszeitpunkt im Signal an. Hierbei ist die Verschiebung so durchzuführen, daß der neue, durch die Verschiebung gewonnene Anfallszeitpunkt nicht in einem peri-iktalen Signalbereich liegt.
- Für die Erzeugung von Measure Profile Surrogaten, werden Signaleile, die vor einem bekannten Anfall (prä-iktal) liegen, aus dem Signal entnommen und mit einem anderen Signalteil, das aus einem anfallsfreien (inter-iktalen) Bereich desselben Signals stammt, vertauscht (siehe Bild 7.3). An diesen Austausch sind Randbedingungen wie der Erhalt der Amplitudenverteilungsdichte und Eigenschaften der Autokorrelation geknüpft. Die Länge des Signalabschnittes, welcher ausgetauscht wird, hängt von der zu untersuchenden prä-iktalen Zeitspanne ab und kann variieren.

Hierfür wird Bezug auf die tatsächlichen Zeitpunkte der Anfänge der Anfälle, die durch Experten der Klinik für Epileptologie in Bonn ermittelt wurden, genommen. In beiden Fällen sind Randbedingungen zu berücksichtigen. Für Seizure Time Surrogate beziehen sich die Einschränkungen darauf, daß die zufällig erzeugten Anfallszeitpunkte nicht in ein Signalsegment fallen dürfen, welches einen Anfall enthält (iktaler Bereich), bzw. in einen prä-iktalen bzw. post-iktalen Signalbereich. Es muß also sichergestellt sein, daß Vertauschungen nur in einen inter-iktalen Signalbereich hinein erfolgen. Für Measure Profile Surrogaten dürfen Signalsegmente vor einem Anfall nicht mit Signalelementen, die zu kurz hinter – post-iktaler Signalbereich – einem Anfall oder auch zu kurz vor einem Anfall – prä-iktaler Signalbereich – liegen, vertauscht werden, d.h. nur Signalelemente aus den sogenannten inter-iktalen Signalbereichen kommen für solche Vertauschungen in Frage. Weiterhin sind Signaleigenschaften wie die statistische Verteilung der Amplituden und der Autokorrelation





**Bild 7.2:** Erzeugung von Surrogaten durch Verschieben von bekannten Anfallszeitpunkten.



**Bild 7.3:** Erzeugung von Surrogaten durch Austausch von Datensegmenten.

zu berücksichtigen. Letztere fließt in die Kostenfunktion während der Optimierung, z.B. mittels Simulated Annealing, mit ein, was für die Erzeugung solcher Surrogate zu rechenintensiven Optimierungen führt. Hierdurch soll gewährleistet werden, daß das zu untersuchende Verfahren in den Surrogaten keine tatsächlichen Anfallszeitpunkte

vorhersagen kann, da die Vorüberlegung, daß von einem tatsächlichen Anfallszeitpunkt im Signal ein prä-iktaler Signalabschnitt vorhanden ist, nun nicht mehr zutrifft. Das Verfahren sollte nun, um  $H_0$  zu widerlegen, keinen tatsächlichen Anfallszeitpunkt mehr in den Surrogaten vorhersagen können, sondern nur noch an den Stellen an denen das prä-iktale Signalelement zu finden ist. Dies entspäche dem Vorhandensein von erkennbaren Merkmalen im prä-iktalen Signalsegmenten. Für die Surrogate die durch Verschieben der tatsächlichen Anfallszeitpunkte auf neue hypothetische Zeitpunkte ohne das Signal zu verändern, erzeugt werden, ergibt sich analog, daß, wenn das zu untersuchende Verfahren an diesen neuen Zeitpunkten Anfälle finden sollte, dies nicht am Vorhandensein von prä-iktalen Vorboten liegen kann, da dort keine Anfälle und auch keine prä-iktalen Signalsegmente zu finden sind (Bestätigung von  $H_0$ ). Werden dennoch Anfälle an den tatsächlichen Anfallszeitpunkten vorhergesagt, würde die Nullhypothese, daß das Verfahren, basierend auf prä-iktalen Vorboten, Anfälle vorhersagen kann, bestätigt und somit  $H_0$  widerlegt werden, sodaß die Gegenhypothese  $H_1$ , daß das Verfahren geeignet ist, in Betracht gezogen werden muß. D.h. wenn das Verfahren zur Vorhersage von epileptischen Anfällen geeignet ist, erhöhen sich die False Positive und False Negative Bewertungen der ROC-Analyse für die Nullhypothese. In diesem Fall sollten die generierten Surrogate zu schlechteren Ergebnissen in der ROC-Analyse führen und die Nullhypothese nicht gestützt werden. Es wird allerdings nicht ausgeschlossen, daß ein Verfahren dennoch die Anfallszeitpunkte richtig prädiziert, allerdings deutet dies darauf hin, daß die prä-iktalen Signalbereiche dafür nicht benötigt werden. Im allgemeinen ist die Zeitspanne des prä-iktalen Signalbereichs nicht genau bekannt, was zu einer möglichen Fehlerquelle bei der Erzeugung von Surrogaten führen kann. Ein möglichst großer Abstand von tatsächlichen Anfallszeitpunkten muß daher mit berücksichtigt werden. Eine weitere Größe, die von der Zeitspanne des prä-iktalen Anfallsverlaufs abhängt, in der also ein Anfall vorhergesagt werden kann, ist der sogenannte Vorhersagehorizont  $H$ . Der Vorhersagehorizont ist die Zeitspanne nach einer

Anfallsvorhersage, in der ein kommender Anfall erwartet wird. Zum einen muß diese Zeitspanne lang genug sein, damit der Betroffene genügend Zeit hat, reagieren zu können, um gefährliche Situationen zu vermeiden. Zum anderen sollte sie aber auch nicht zu lang sein, da die betroffene Person dann zu lange auf das Eintreten des Anfalls warten muß und erst nach Ablauf der Vorhersagezeit feststeht, ob es sich tatsächlich um einen Anfall oder um einen Fehllarm handelte.

### 7.1.1 EEG-Signale und epileptische Anfallsvorhersage

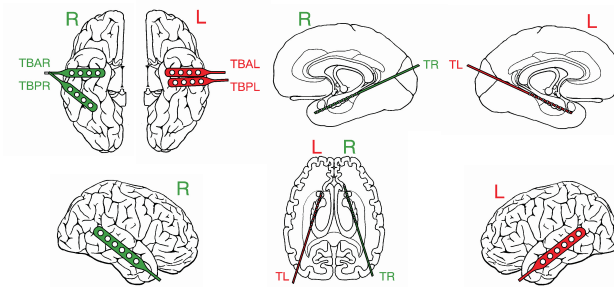
Das zentrale Nervensystem von Menschen arbeitet nachweislich auf der Basis elektrischer Impulse, die gesendet und empfangen werden. Mittels Elektroden auf der Kopfhaut können derartige Signale als Summensignale der Hirnaktivität aufgenommen werden. Es handelt sich hierbei um Signale die mit einer Dipolcharakteristik in Verbindung gebracht werden können. Die Entstehung ist zwar auf die Aktivität von Neuronen zurückzuführen, deren Einzelaktivitäten oder synchrone Aktivitäten in den sogenannten Zellensembles im Signal sind aber nicht mehr eindeutig zu erkennen. Dennoch können mit solchen Aufnahmen über den Gesamtzustand von Hirnarealen Aussagen getroffen werden[77, 78]. Diese Signale werden allgemein als EEG bezeichnet.

Für die chirurgische Behandlung von Patienten, die unter epileptischen Anfällen leiden, reicht die mit EEG-Signalen erreichbare räumliche Auflösung und Signalqualität häufig nicht aus; deshalb müssen in solchen Fällen Flächen- bzw. Tiefenelektroden für eine begrenzte Zeit subdural – unter die Schädeldecke unter die harte Hirnhaut – implantiert werden, um den Anfallsherd – Fokus – genauer bestimmen zu können. Die Signale dieser Meßmethoden werden als ECoG (**Elektrocorticogramm**) bzw. SEEG (**Stereoelektroencephalogramm**) bezeichnet. Im Folgenden werden solche Hirnstromaufnahmen der

Einfachheit halber ebenfalls mit dem Begriff „EEG-Signal“ bezeichnet. Sie bieten eine höhere räumliche Auflösung als Aufzeichnungen des herkömmlichen EEG und erlauben so eine genauere Bestimmung des Fokus. Weiterhin zeigen sie in ihrem zeitlichen Verlauf eine deutliche Veränderung bei Eintreten eines Anfalls – iktaler Signalbereich – verglichen mit dem Signal aus einem anfallsfreien – inter-iktalen – Bereich.

### 7.1.2 Datenbasis

Für die Untersuchungen standen Daten aus EEG-Ableitungen von 12 Epilepsie Patienten des Uniklinikums Bonn zur Verfügung. Die Aufnahmen wurden im Rahmen von Untersuchungen bei Patienten mit fokaler Epilepsien mittels subduraler Oberflächen- oder Tiefenelektroden, die auf der Gehirnoberfläche<sup>2</sup> bzw. Tiefenelektroden, die seitlich in das Hirn implantiert werden, gewonnen, wie es beispielhaft im Implantationsschema in Bild 7.4 gezeigt wird. Solche Signale erlauben Ex-

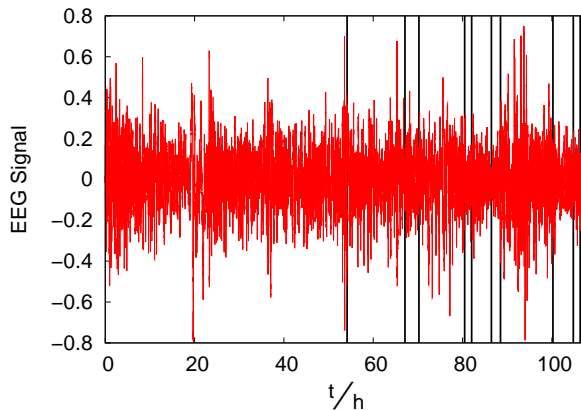


**Bild 7.4:** Implantationsschema von Elektroden. Jede Elektrode hat mehrere Kanäle.

perten bei Eintreten eines Anfalls diesen zu erkennen und ermöglichen im Bedarfsfall eine genaue Signalanalyse. Bei den Datensätzen

<sup>2</sup>Bei subdural, unter die harte Hirnhaut, implantierte Elektroden .

handelt es sich um Langzeitaufnahmen von mehreren Tagen mit bis zu 48 Signalkanälen – entsprechend der Anzahl aller Kontakte aller implantierter Multikontaktelektroden – und sind aufgrund des großen Datenvolumens auf mehrere Dateien aufgeteilt. Im Gegensatz zu Kurzzaufnahmen, die meist nur ein paar Minuten bis zu einer Stunde andauern, kann bei Langzeitaufnahmen das Auftreten von Anfällen z.B. durch Absetzen von Medikamenten unter klinischer Beobachtung, provoziert und aufgezeichnet werden. Ein Beispiel solch eines Signales ist in Bild 7.5 gezeigt. Der Schwerpunkt



**Bild 7.5:** Beispiel eines Langzeit EEG-Signals (—) über 108 Stunden. Anfälle sind durch senkrechte Linien markiert( | ).

wird hier auf die Entwicklung, Implementierung und Untersuchung von CNN-Algorithmen gelegt. Einer der hier vorgestellten Algorithmen wurde bereits in anderen Arbeiten untersucht ([69, 70]) und seine Tauglichkeit für die Prädiktion von epileptischen Anfällen gezeigt, so daß hier die Untersuchung seiner Umsetzung auf verfügbare schaltungstechnische Realisierungen von CNN im Vordergrund steht. Für die Verarbeitung auf solchen CNN Schaltungen ist eine Vorverarbeitung der Daten aus zwei Gründen notwendig:

- Die Anzahl Zellen einer schaltungstechnischen Realisierung ist stark begrenzt, weshalb das Signal in kleine Teile zerlegt werden muß und
- die Daten für die Verarbeitung auf dem EyeRIS-System an dessen Auflösungsgenauigkeit angepaßt werden müssen.

Weiterhin ist davon auszugehen, daß EEG-Signale nicht stationär sind und deshalb nur in kleinen Zeitabschnitten Stationarität angenommen werden kann, daher ist eine, wie im ersten Punkt genannte Vorgehensweise, auch eine für Simulationen sinnvolle Vorverarbeitung. Hierbei werden die Daten in Signalabschnitte, Segmente von 10 Sekunden Dauer, zerlegt, in denen das Signal als quasistationär angenommen wird. Da das EyeRIS-System für die Bildverarbeitung ausgelegt ist, lag es nahe, die als Zeitreihen vorliegenden Datenssegmente in eine zweidimensionale Darstellung umzuwandeln. Während der Vorverarbeitung ist es nötig, die Daten auf 8 bit Genauigkeit umzurechnen, da dies durch die Auflösung der DA/AD-Wandler des EyeRIS-Systems bedingt wird. Dieses führt zu Informationsverlusten, was die Prädiktion von epileptischen Anfällen mit dem hier verwendeten EyeRIS-System sehr erschwert. Aufgrund dessen, daß die Übertragungszeit von Datensätzen auf das EyeRIS-System und wieder zurück auf einen das System ansteuernden Rechner wesentlich zur Gesamtlaufzeit beiträgt, ist eine umfassende Untersuchung aller Daten im Rahmen dieser Arbeit nicht möglich. Deshalb werden die Untersuchungen nur für ein Teil aller 48 Kanäle eines Patienten für die folgenden Betrachtungen herangezogen. Es handelt sich bei dem Patienten um einen jungen Mann der im Klinikum für Epileptologie in Bonn behandelt wurde und dessen EEG im Datensatz Nummer 6 in 16 Einzeldateien, von Datei A bis P, abgelegt sind. Für die untersuchten Elektroden wurde das Langzeit EEG für die gesamten 108 Stunden verwendet. Hierbei wurde das Signal in Segmente von je 10 Sekunden zerlegt und diese Zeitspanne als quasistationär angenommen.

### 7.1.3 Prädiktion mit DT-CNN

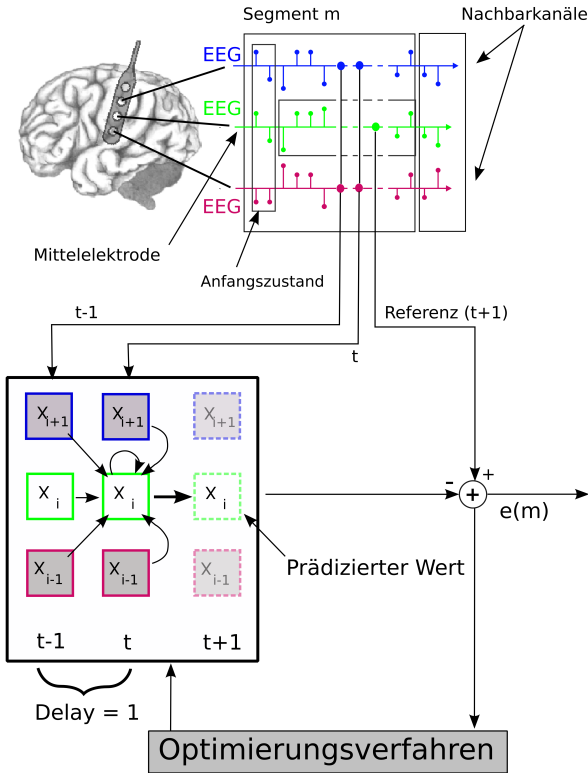
Eines der in dieser Arbeit untersuchten Verfahren zur Prädiktion epileptischer Anfälle basiert auf der Optimierung von Prädiktorkoeffizienten für als stationär angenommene kurze Segmente des EEG-Signals. Für jedes Segment wird ein Prädiktionsfehler und damit ein neues Zeitsignal mit den Prädiktionsfehlern für jedes konsekutiv folgende Segment gebildet. Zur Vorhersage eines epileptischen Anfalls wird auf eine signifikante Änderung im Prädiktionsfehler geachtet. Das Verfahren ist ein multi-variates Verfahren, da mehrere Kanäle in die Prädiktion einbezogen werden (wie in Bild 7.6 illustriert). Um eine Portierung des Algorithmus auf das EyeRIS-System für die gegebene Epilepsieproblematik möglich zu machen, bedarf es einer Anpassung der CNN Zustandsgleichung (2.1) bzw. (2.17). Da der analoge ACE16kv2-Chip die direkte Verwendung von Time-Delay-Templates nicht unterstützt, mußte eine Zuordnung von Zeitpunkten auf bestimmte <sup>3</sup> Zellen des Netzwerks vorgenommen werden. Hierbei werden die Signalwerte der einzelnen Zeitpunkte linear in einer Zeile und die Werte der drei im Algorithmus verwendeten Kanäle, entsprechend Bild 7.7, als nebeneinander liegende Zeilen auf dem CNN angeordnet. Hierbei werden die zeitlich aufeinander folgenden Werte des Signals räumlich auf die Pixel in horizontaler Richtung nebeneinander angeordnet. Die Werte der Nachbarelektroden werden parallel dazu angeordnet. Für die Prädiktion des Signals wird nun eine Template

$$\mathbf{W}x_1 = \begin{pmatrix} w_1 & w_4 & 0 \\ w_2 & w_5 & 0 \\ w_3 & w_6 & 0 \end{pmatrix}, \quad (7.1)$$

erstellt und die  $w_i$  optimiert. Eine Zelle wird also nur durch Nachbarzellen die links, also zeitlich bei  $t - 1$ , sowie Zellen die oben und

---

<sup>3</sup>Diese Zellen entsprechen damit den aus der zeitdiskreten Filtertheorie bekannten Zustandsspeichern.

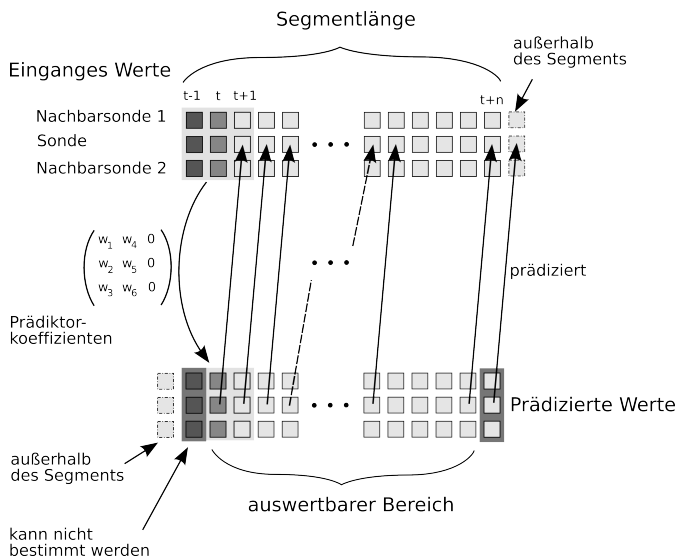


**Bild 7.6:** Exemplarisch sind hier die Signale dreier benachbarter Kanäle, sowie ihre Verarbeitung im Prädiktor skizziert.

unten und sich selbst, also zum Zeitpunkt  $t$ , beeinflusst. Der prädizierte Wert steht nach Anwendung dieser Template daher an der Stelle der Zelle selber. Für das Optimierungsverfahren ist es nun nötig, diesen Wert mit dem tatsächlichen, bei  $t + 1$  vorhandenen Wert zu vergleichen, um den Fehler zu errechnen und entsprechend die Gewichtungen anzupassen, siehe Bild 7.7. Da die Abbildung eines zeit-



lichen Signals auf eine räumliche Struktur mit begrenzter Ausdehnung, wie es das EyeRIS-System darstellt, mit virtuellen Zellen an den Rändern arbeitet, müssen Randeffekte berücksichtigt werden, was die Anzahl räumlich eingebetteter Zeitschritte begrenzt. Da das EyeRIS-System nur eine  $3 \times 3$  Template unterstützt, ist die multivariate Prädiktion mit Zeitverzögerung um nur einen Abtastwert auf dem Chip-System realisierbar; Varianten des Algorithmus mit längerer Verzögerung sind auf derzeit verfügbarer CNN-Hardware nicht realisierbar. Die Güte der Prädiktion wird über den RRMSE nach



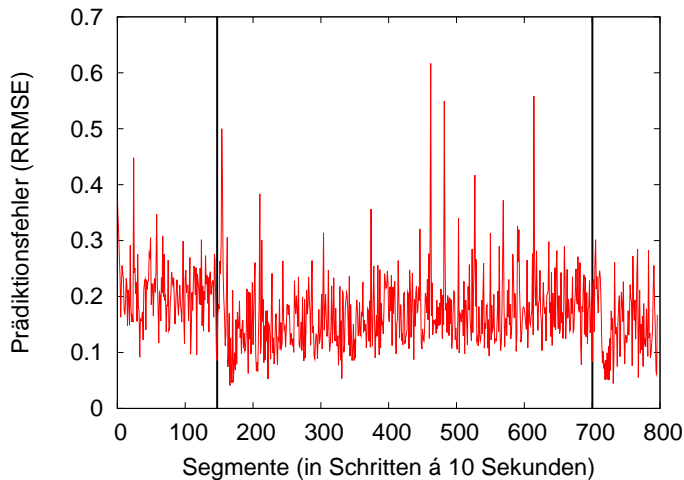
**Bild 7.7:** Umsetzung des Delay auf dem verwendeten Chip-System.

Gl. (5.1), bestimmt. Die Qualität und Verlässlichkeit wurde mit der Methode der Receiver-Operating-Characteristik [43, 44, 79, 80, 45] und mit Surrogaten untersucht, wobei das in [75, 47, 48, 65] beschriebene Vorgehen angewandt wurde. Hierbei wurde das Zeitsignal des Prädiktionsfehlers als auch Surrogate eines solchen, die durch Verschieben des Anfallszeitpunktes – Seizure Time Surrogate – erzeugt

werden, mittels Bestimmung und Analyse von AUC-Werten untersucht und die Signifikanz bestimmt. Weiterhin wurden umfangreiche Analyse von Optimierungsalgorithmen bezüglich der Prädiktionsgüte vorgenommen. Untersucht wurden:

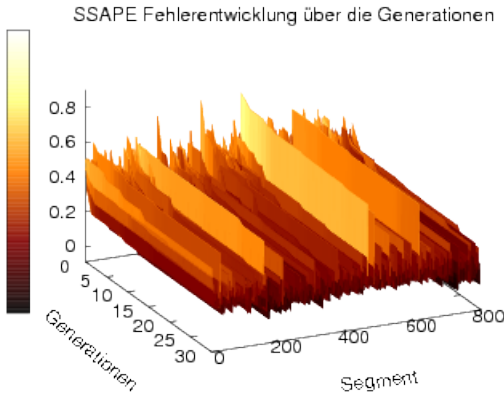
- Anwendung von Optimierungsverfahren auf einem PC mit doppelter Gleitkommagenauigkeit der Daten und der Template-darstellung:
  - Broyden–Fletcher–Goldfarb–Shannon (BFGS)[31]
  - Differential Evolution (DE)[41][42]
  - Symmetry-Supporting-Annealed-Parameter-Evolution (SSAPE)
  - Simulated Annealing entsprechend [31]
- Anwendung von Optimierungsverfahren auf einem PC mit 8 bit Genauigkeit zwecks Vergleich mit den Ergebnissen der Untersuchungen mit dem EyeRIS 1.1 Chip-System:
  - Differential Evolution (DE)
  - Symmetry-Supporting-Annealed-Parameter-Evolution (SSAPE)
- Auf dem EyeRIS 1.1 System:
  - Differential Evolution (DE)
  - Symmetry-Supporting-Annealed-Parameter-Evolution (SSAPE).

In Voruntersuchungen wurde das Laufzeitverhalten der Optimierungsalgorithmen betrachtet, um die benötigten Iterationsschritte bis zum Erreichen eines möglichst kleinen Prädiktionsfehlers (RRMSE) für die nicht-deterministischen Optimierungsverfahren zu erhalten. Es zeigte sich, daß schon nach wenigen Iterationsschritten Prädiktionsfehler erreicht wurden, die sich bei weiteren Iterationen (Generationen) nur noch wenig verbesserten, was im Falle von SSAPE zu einer Wahl von 30 Iterationen führte (siehe Bild 7.8).



**Bild 7.8:** RRMSE für SSAPE nach 30 Iterationen (Generationen) für ein EEG-Signal welches zwei Anfälle enthält. Aufgetragen ist der Prädiktionsfehler von Segmenten je 10 Sekunden gegen die Segmentnummer für Datei P des Gesamtdatensatzes 6.

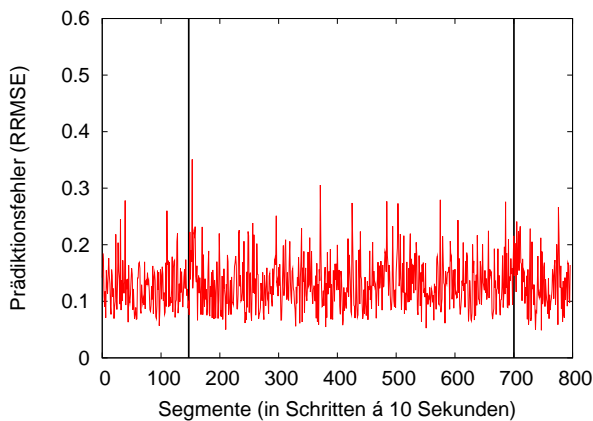
Die Veränderung des Fehlers über die jeweils ausgeführten Iterationen – Generationen – ist exemplarisch in 7.9 gezeigt. Der in der 30sten Generation erreichte Fehler entspricht dem in Bild 7.8 gezeigten. Der in dieser Arbeit vorgestellte SSAPE-Algorithmus zeigte sich in allen vorgenommenen Untersuchungen als mindestens gleichwertig zu den anderen untersuchten Optimierungsverfahren. Vor allem auf dem EyeRIS-System erwies der Algorithmus sich in seiner Ausführungsgeschwindigkeit, mit bis zu 7 Datensegmenten pro Minute als der schnellere Algorithmus verglichen mit dem Differential Evolution Algorithmus, der nur maximal 5 Datensätze in der Minute verarbeitete. Unter Berücksichtigung, daß das EyeRIS-System ab und an neu gestartet werden mußte, ergab sich für eine Elektrode pro untersuchtem Optimierungsverfahren eine Rechenzeit von circa einer Woche, die um die Wiederholbarkeit zu überprüfen, mehr-



**Bild 7.9:** Entwicklung des Prädiktionsfehlers mit SSAPE über 30 Generationen. Aufgetragen ist der Prädiktionsfehler von Segmenten je 10 Sekunden eines EEG-Signals für Datei P mit zwei Anfällen über 30 Iterationen (Generationen).

fach durchgeführt wurden. Ein Vergleich mit dem Simulated Annealing und dem BFGS Verfahren wurde auf dem Chip-System nicht durchgeführt, obgleich diese Verfahren auf einem PC mit doppelter Gleitkommagenauigkeit mit die besten Ergebnisse lieferten (A.28 und A.29). Dies ist damit begründet, daß sich für das Simulated Annealing bereits in einer Voruntersuchung (Simulationen auf einem PC) für einen Auflösungsgrad von 8 bit ein Laufzeitbedarf von gut 6 Tagen zeigte, wobei hierbei bereits mehrere Rechner parallel einzelne Datensegmente berechneten. Für das BFGS stellte sich heraus, daß aufgrund der schaltungstechnisch bedingten Abweichungen – geringer Auflösungsgrad von 8 bit, Verluste während der Verarbeitung im analogen Chip –, die sichere und genaue Bestimmung des Gradienten der Fehlerfunktion, der für die Optimierung notwendig ist, nicht möglich war und daher das BFGS-Verfahren abbrach.

In weiteren Voruntersuchungen stellte sich heraus, daß auf dem Eye-RIS System zufällige Abweichungen in den Ergebnissen auftraten, die sich auf die Reproduzierbarkeit auswirkten. Um diesem entgegen zu wirken und dennoch eine aussagekräftige Prädiktion zu erzielen, wurden Parametersätze mehrfach angewandt und ein mittlerer Fehler bestimmt. Hierbei wurde untersucht, inwieweit sich die einfache Anwendung von einer mehrfachen Anwendung und Mittelung der erhaltenen Prädiktionsfehler auswirkt.



**Bild 7.10:** RRMSE für SSAPE für ein EEG-Signal welches zwei Anfälle enthält (senkrechte Markierungen). Aufgetragen ist der Prädiktionsfehler von Segmenten je 10 Sekunden gegen die Segmentnummer. Bei dieser Untersuchung wurden die Berechnungen für jeden Parametersatz fünfmal ausgeführt und der mittlere Fehler der fünf Berechnungen gebildet.

Wie in Bild 7.10 gezeigt, reduziert sich der Gesamtfehler im Vergleich zu Bild 7.8. Die Mittelung der Fehlerwerte führt allerdings zu schwächer ausgeprägten Signalverläufen bei den Anfallszeitpunkten. Weiterhin ist durch die fünfmalige Fehlerbestimmung die Laufzeit einer solchen Berechnung länger, was einen nicht vertretbaren Zeitbe-

darf von mehreren Wochen für fünffach wiederholte Berechnungen eines Kanals bedeutete. Es wurde festgestellt, daß keine wesentliche Verbesserung des Verfahrens resultierte, weshalb in späteren Untersuchungen auf die wiederholte Anwendung von Templates und Mittelung des Fehlers verzichtet wurde.

#### **7.1.4 Ergebnisse der Prädiktion durch Anwendung von Optimierungsverfahren**

Die auf dem EyeRIS-System erhaltenen und die durch Simulation auf einem PC gewonnenen Ergebnisse, die mit einer auf 8 bit reduzierten Genauigkeit berechnet wurden, werden hier exemplarisch verglichen. Die Reduzierung der Berechnungsgenauigkeit wurde durchgeführt, um eine bessere Vergleichbarkeit der Ergebnisse zu erhalten. Es zeigte sich, daß die reduzierte Genauigkeit des EyeRIS-Systems in den Berechnungen starken Einfluß auf die Ergebnisse hat und diese auf Grund der Surrogatanalyse als nicht signifikant betrachtet werden müssen. Ergebnisse der Simulationen mit reduzierter Genauigkeit auf einem PC zeigten etwas bessere Ergebnisse als die Berechnungen auf dem Chip. In einem Fall ergab sich für eine Simulation mit 8 bit Genauigkeit sogar ein besseres Ergebnis als die vergleichbare Simulation mit Gleitkommazahlberechnung, was aber als Ausnahme zu werten ist und was in den Bildern unter A.25 sowie A.27 gezeigt ist.

Das EyeRIS-System mit seinem analogen ACE16kv2 Chip zeigt jedoch noch eine weitere Ungenauigkeit da seine AD/DA Wandler zwar mit 8 bit Genauigkeit arbeiten, die tatsächliche Genauigkeit, wie die Untersuchungen in Abschnitt 3.2 zeigen, allerdings mit 5 bit anzunehmen ist. Einige der Ergebnisse sind in den Bildern 7.11 und 7.12 dargestellt. Weiterhin zeigte sich, daß die Ergebnisse stark vom verwendeten Vorhersagehorizont abhängen. Wurde eine feste Zeitspanne vorgegeben, so kam es vor, daß manche Ergebnisse keine si-

gnifikanten Vorhersagen eines Anfalls erlaubten, wie in Tabelle 7.1 aufgeführt ist. Für einige der Berechnungen mit doppelter Gleitkommaengenauigkeit konnten signifikante Ergebnisse erzielt werden, wenn der Vorhersagehorizont  $H$  entsprechend gewählt wurde, wie in Tabelle 7.2 aufgeführt ist. Weitere Ergebnisse sind in Anhang A gezeigt.

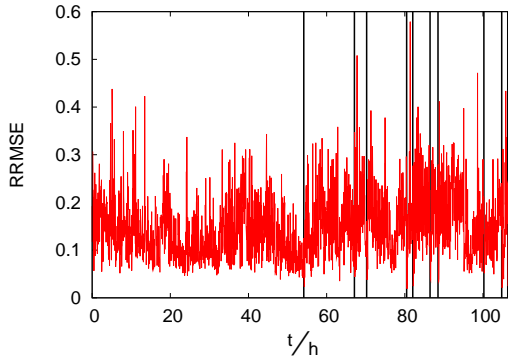
TLI06	0.609	0.599	<b>0.655</b>	0.651	0.645	0.651	0.643	<b>0.724</b>
TLI06	0.535	0.536	0.612	0.615	-	-	-	-
TBPR4	0.562	0.714	0.649	0.646	-	-	-	-
TBPI4	0.577	0.611	0.681	0.685	-	-	-	-
TBAR4	0.637	0.696	0.581	0.580	-	-	-	-
DE	SSAPE	DE	SSSPE	DE	SSAPE	BFGS	Simulated Annealing	
Kanal	Chip	PC 8 bit	PC double					

**Table 7.1:** Liste der AUC-Werte für die untersuchten Kanäle über die Gesamtzeit der Aufzeichnung von Patient 6 mit konstantem Prädiktionshorizont von 120 Minuten. Signifikante Ergebnisse sind fett dargestellt.

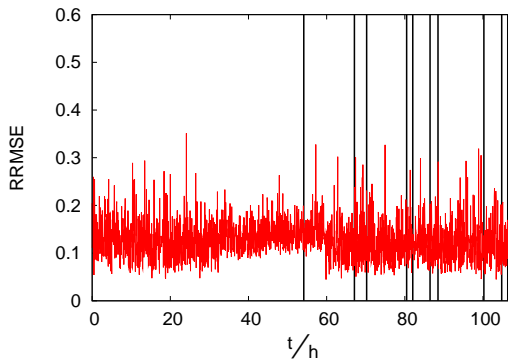
TLI06	0.596 *	0.599 *	<b>0.629</b> ⊕	<b>0.645</b> ●	<b>0.612</b> ⊗	<b>0.646</b> ⊕	<b>0.623</b> *	<b>0.718</b> ○
TLI06	0.523 *	0.513 *	<b>0.608</b> *	0.594 *	-	-	-	-
TBPR4	0.557 ●	<b>0.698</b> ⊖	<b>0.631</b> ●	0.628 ●	-	-	-	-
TBPI4	0.577 ⊖	0.544 *	<b>0.651</b> ○	<b>0.653</b> ●	-	-	-	-
TBAR4	0.637 ⊙	<b>0.695</b> *	0.534 *	0.579 *	-	-	-	-
DE	SSAPE	DE	SSAPE	DE	SSAPE	BFGS	Simulated Annealing	
Kanal	Chip	PC 8 bit	PC double					

**Table 7.2:** Liste der AUC-Werte für die untersuchten Kanäle über die Gesamtzeit der Aufzeichnung von Patient 6 mit Prädiktionshorizont für beste AUC Werte 60 ●, 80 ○, 90 ⊕, 100 ⊗, 120 ⊙, 150 \*, 180 ⊖, 210 ⊙ und 360 \* Minuten. Signifikante Ergebnisse sind fett dargestellt.



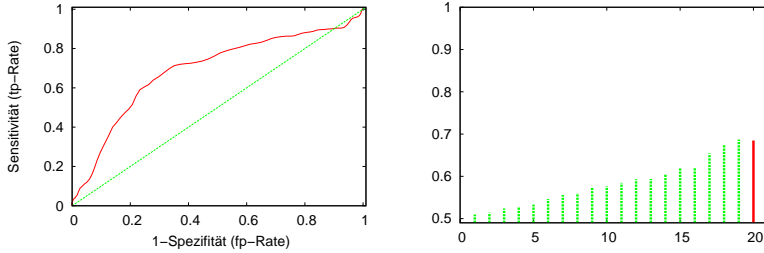


(a)

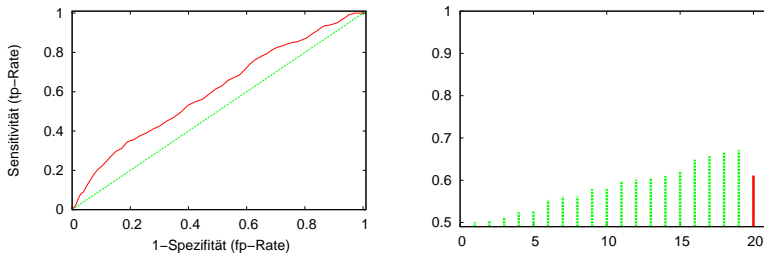


(b)

**Bild 7.11:** Typische Ergebnisse für die Fehlerwerte als Funktion der Zeit (—). Ein Anfallsbeginn ist jeweils durch ( | ) gekennzeichnet. (a) Prädiktionsfehler  $e_{DT-CNN}^{8bit}(t)$  für DT-CNN simuliert mit 8 bit Genauigkeit für den Kanal TBPL4, (b)  $e_{DT-CNN}^{EyeRIS}(t)$  wie vorhergehend aber auf dem EyeRIS 1.1 System gerechnet. Die Prädiktionsfehler liegen in der gleichen Größenordnung unterscheiden sich aber.



(a)



(b)

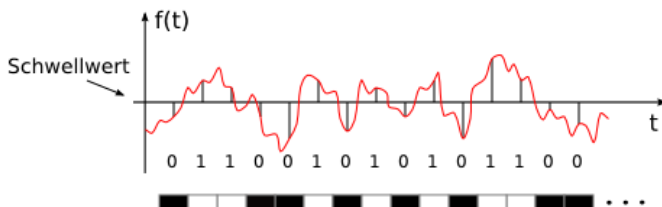
**Bild 7.12:** Links: ROC-Kurven; rechts: dazugehörige AUC-Werte, Linien bei 20 (—), und Ergebnisse der Surrogate, Linien bei 1– bis 19 (—) für EEG-Signal: (a)  $e_{DT-CNN}^{8\text{bit}}$ ,  $H=210\text{min}$ , (b)  $e_{DT-CNN}^{EyeRIS}$ ,  $H=60\text{min}$ , für Kanal TBPL4. Die auf dem CNN Chip berechneten Ergebnisse, erscheinen deutlich schlechter gegenüber denen auf einem PC mit auf 8 bit reduzierter Genauigkeit berechneten. Es sind Ergebnisse für die Vorhersagehorizonte gezeigt, für die die besten AUC-Werte erlangt wurden. (siehe auch Tabellen 7.1 und 7.2).

### 7.1.5 Prädiktion durch Mustererkennung und Symbolisierung

In diesem Abschnitt soll der Ansatz untersucht werden, inwiefern die in einem EEG Signal angenommenen prä-iktalen Vorboten eines epileptischen Anfalls, durch Symbolisierung und anschließender Analyse der Auftrittshäufigkeit von Symbolen bzw. Mustern, in der Symbolabfolge detektiert werden kann. Ein Symbol ist hierbei ein Element einer endlichen oder unendlichen Menge von Informationsträgern, einem Alphabet. Über die Auftrittshäufigkeit der Symbole ergibt sich durch die Shannon-Entropie deren Informationsgehalt. Symbole sind in dem hier untersuchten Kontext als Muster von schwarzen und weißen Pixeln zu verstehen und werden aus einem zeitlich diskretisierten, abgetasteten Signal erzeugt. Zur Symbolisierung bietet sich z.B. unter der Annahme, daß das Signal mittelwertbefreit ist, o.B.d.A. eine Schwellwertfunktion der Form

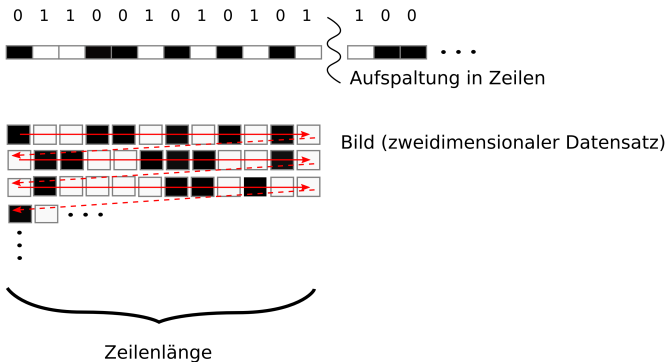
$$f_o(t) = \begin{cases} 1 & f_i(t) > S \\ 0 & f_i(t) \leq S \end{cases} \quad (7.2)$$

mit Schwellwert  $S$ , Eingangssignal  $f_i$  und Ausgangssignal  $f_o$  an [81, 82]. Für z.B.  $S = 0$  kann hierdurch das abgetastete Signal in eine binäre Abfolge von 0 und 1, wie es in Bild 7.13 illustriert ist, umgewandelt werden. Ein Verfahren, welches sich auf die Auftrittshäufig-



**Bild 7.13:** Symbolisierung am Beispiel eines Schwellwertes.

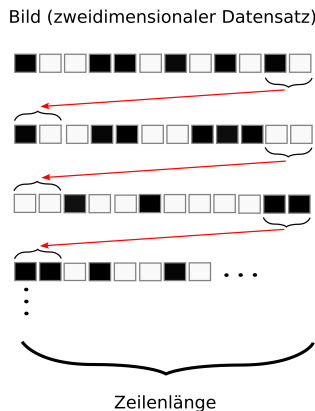
keit von Mustern im symbolisierten Signal konzentriert, wurde schon in [81] und [83] vorgestellt und in [84] erweitert. Die hier durchgeführten Untersuchungen konzentrieren sich auf die Implementierung eines solchen Verfahrens auf dem EyeRIS-System. Sowohl die Schwellersymbolisierung als auch die Mustererkennung sind Operationen, die sich mit CNN besonders effizient durchführen lassen. Mittels der Threshold-Template gemäß Abschnitt 2.2.4 kann die Schwellersymbolisierung durchgeführt werden. Das Verfahren in Kapitel 4 realisiert die Mustererkennung. Hierzu ist es allerdings notwendig, die zu untersuchenden Datensätze in zweidimensionale Datensätze – Bilder – einzubetten (siehe Bild 7.14).



**Bild 7.14:** Anordnung von Signalwerten in einer zweidimensionalen Datenstruktur – einem Bild.

Die Größe der zu bearbeitenden Bilder ist hierbei auf die Größe des CNN-Netzwerkes auf dem EyeRIS-System von maximal  $128 \times 128$  Zellen beschränkt. Daher müssen Signale, die länger als 128 Abtastwerte sind, umgebogen werden. An den Rändern des Netzes führen die Zellen des Netzwerkes, die noch im Netzwerk sind, aber virtuelle Randzellen in ihrer Nachbarschaft haben, zu Detektionen von Mustern, die nicht im ursprünglichen Signal enthalten sind. Hierdurch

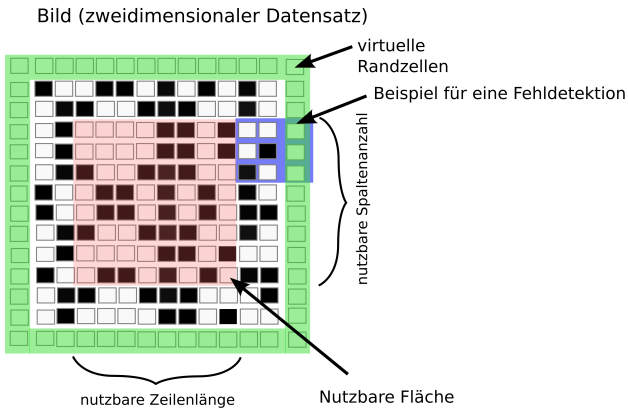
reduziert sich die verwendbare Spaltenanzahl und Zeilenanzahl. Die Fehldetektionen werden durch Beschränken der Auszählung auf Zellen, deren Nachbarschaft komplett im Netzwerk liegt, also Zellen die nicht an virtuelle Randzellen angrenzen, bei der Auswertung bereinigt. Werte aus dem binarisierten Signal, die an der Stelle der Aufspaltung in Zeilen vorhanden sind, würde man so aber nicht berücksichtigen. Um dennoch alle Werte aus dem binarisierten Signal zu berücksichtigen, werden die Signalwerte, die den letzten zwei Zellen einer Zeile zugeordnet sind, also an der Stelle der Aufspaltung liegen, am Anfang der folgenden Zeile wiederholt, damit Muster, die an der Stelle der Aufspaltung vorhanden sind, mit berücksichtigt werden können, wie in Bild 7.15 gezeigt. Hierdurch verschiebt sich das eingebettete Signal je Zeile um zwei Signalwerte.



**Bild 7.15:** Anordnung eines symbolisierten Signals unter Berücksichtigung von Randeffekten in einem 2D-Datensatz (vergleiche hierzu auch Bild 4.1).

Weiterhin muß berücksichtigt werden, daß das Mustererkennungsverfahren aus Kapitel 4 mit Verschiebeoperationen arbeitet. Hierdurch

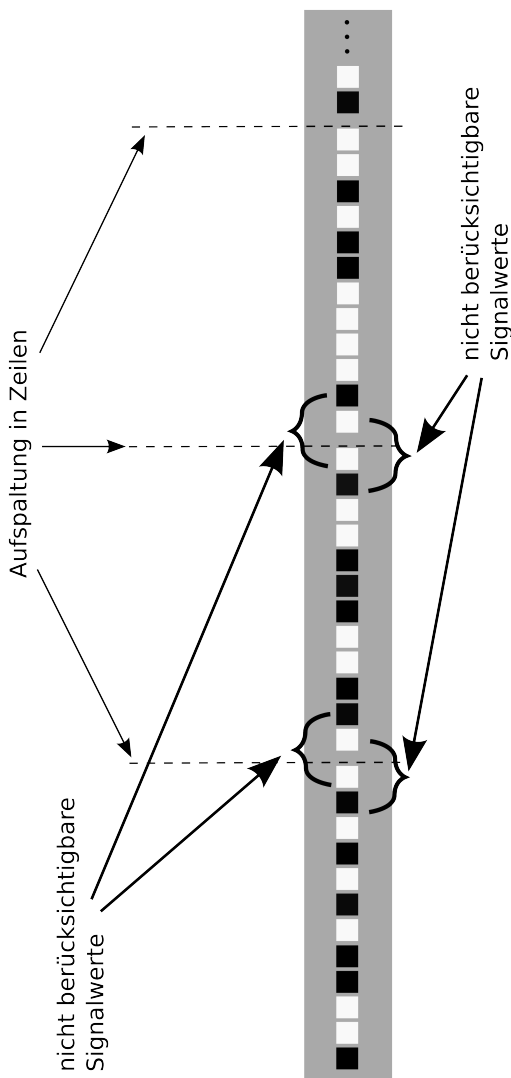
werden an den Rändern des CNN-Netzwerks die Zustandswerte der virtuellen Randzellen, je nach Richtung der Verschiebung, in das Netzwerk hineinverschoben und werden somit Zustandswerte realer Zellen. Dadurch entstehen dort Bereiche, die sich ebenfalls nicht zur Detektion von Mustern im Signal eignen. Deshalb ist die nutzbare Zeilenanzahl und Zeilenlänge an jeder Seite noch einmal um je ein Pixel kleiner, wie in Bild 7.16 schematisch für ein  $12 \times 12$  Netzwerk dargestellt. Für das EyeRIS System bedeutet dies eine nutzbare Pixelanzahl von  $124 \times 124$  Pixeln.



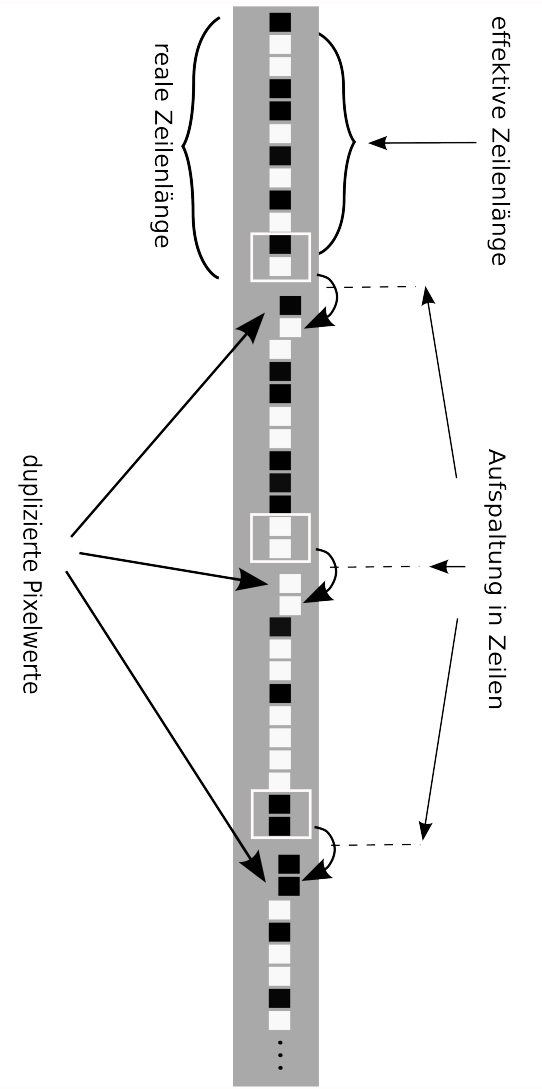
**Bild 7.16:** Schematische Darstellung für die in einer Mustererkennung nutzbaren Fläche, gezeigt für ein  $12 \times 12$  Netzwerk. Gezeigt ist, wie Randeffekte zu einer Reduzierung der für die Mustererkennung nutzbaren Zellenanzahl, respektive Anzahl an Pixeln, führt.

Die Zeilen eines  $3 \times 3$  Musters setzen sich nun aus binarisierten Signalwerten zusammen die aufeinander folgen, während Musterpixel in den Spalten im ursprünglichen Signal um die Länge einer Zeile abzüglich der Länge einer Musterzeile, hier drei Pixel, auseinander liegen, wie die Bilder 7.17, 7.18 und 7.19 veranschaulichen.

Berücksichtigung von Mustern an der Auftrennung zu Zeilen



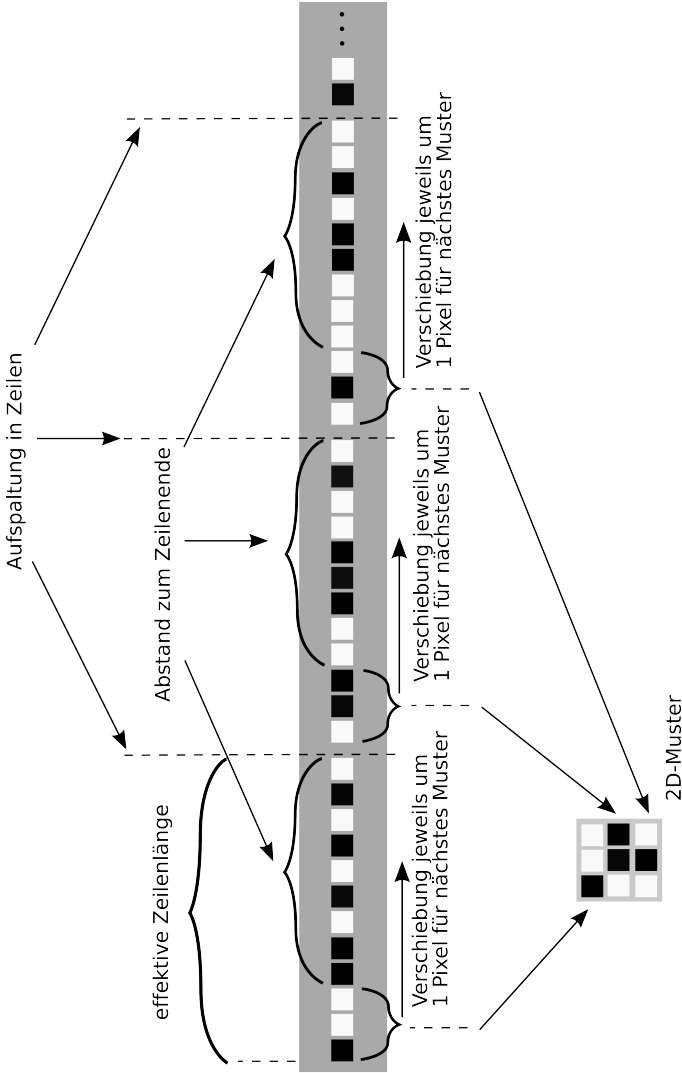
**Bild 7.17:** Einbettung eines Zeitsignals in ein Bild ohne Berücksichtigung der Muster an der Stelle der Aufspaltung.



**Bild 7.18:** Einbettung eines Zeitsignals in ein Bild mit Berücksichtigung der Muster an der Stelle der Aufspaltung.



Muster in einem linearen Datensatz und korrespondierendes Muster in einem zweidimensionalen Datensatz



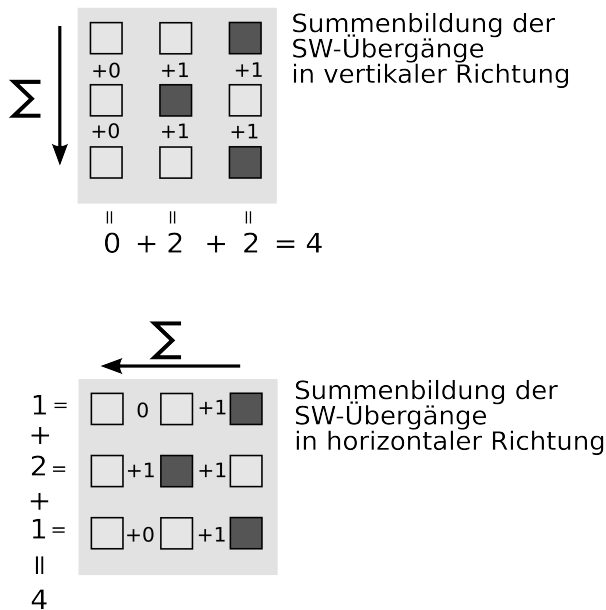
**Bild 7.19:** Muster und Musterdetektion in einem eindimensionalen Signal für ein zweidimensionales Beispielmuster.

### 7.1.6 Statistik der Musterhäufigkeiten

Der Grundgedanke der im folgenden aufgezeigten Methode besteht darin, daß bestimmte Muster bzw. Typen von Mustern vor einem Anfall ihre Auftrittshäufigkeit verändern. Dies könnte dann zur Vorhersage von epileptischen Anfällen genutzt werden. Die Eigenschaften, die ein solches Signal vor allem auszeichnen, sind seine Frequenzanteile, WS-Wechsel, und seine Periodizität der 3 Zeilen des  $3 \times 3$  Musters. Diese Merkmale lassen sich durch Einführen von Musterklassen untersuchen. Durch die Musterbildung mit Hilfe eines Schwellwertes werden im Allgemeinen Durchgänge durch den Schwellwert und im speziellen – bei einem Schwellwert von 0 – Nulldurchgänge in schwarz/weiß-Wechsel umgesetzt. Durch Auszählen der schwarz/weiß-Wechsel (SW-Wechsel) einzelner Zeilen in horizontaler Richtung, d.h. der  $3 \times 1$  Teilmuster eines  $3 \times 3$  Musters, erhält man eine der Frequenz der Nulldurchgänge proportionale Größe. Durch Auszählen der SW-Wechsel einzelner Spalten in vertikaler Richtung, d.h. der  $1 \times 3$  Teilmuster eines  $3 \times 3$  Musters, erhält man eine Größe, welche der Ähnlichkeit der  $3 \times 1$  Abschnitte proportional ist und somit ein Maß für die Wiederholung der  $3 \times 1$  Teilmuster im Gesamtmuster, eine Musterkorrelation, darstellt [85]. Die Zeilen eines solchen Musters liegen im binarisierten, eindimensionalen Signal um die Länge einer Zeile der zweidimensionalen Representation an Abtastwerten abzüglich der Teilmusterlänge  $3 \times 1$  in horizontaler Richtung – im betrachteten Fall also minus 3 – auseinander, vergleiche Bild 7.19.

Durch Zuordnung der Summen der schwarz/weiß-Übergänge (SW-Übergänge) in den Zeilen ergeben sich 7 Musterklassen für die SW-Wechsel, eine Wechselfrequenz. Entsprechend ergeben sich 7 Musterklassen durch die Aufsummierung der SW-Übergänge in den Spalten, welche zur Begutachtung der Ähnlichkeit von Mustern herangezogen werden kann, siehe Bild 7.20, und im folgenden als Musterkorrelation

bezeichnet wird.

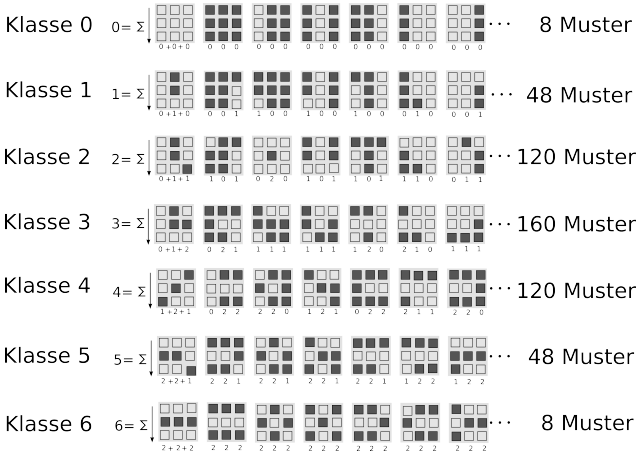


**Bild 7.20:** Bildung der Musterklassen für horizontale und vertikale Aufsummierung.

Einige der möglichen Muster zu den auf diese Weise gebildeten 7 Musterklassen für horizontale Summenbildung sind in Bild 7.21 zu sehen.

Die Musterklassen, die hierdurch erzeugt werden, enthalten unterschiedlich viele Muster. Dies führt dazu, daß einige Musterklassen häufiger detektiert werden als andere, jedoch geben die Musterklassen, die eine besonders geringe Korrelation aufzeigen, deren Spaltensumme also klein und somit die Teilmusterübereinstimmung groß ist (z.B. Musterklassen 0 und 1), einen Hinweis auf ein verändertes

Beispiele für die Musterklassen

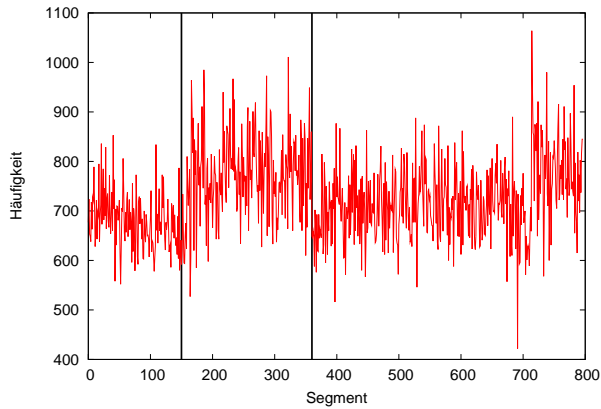


**Bild 7.21:** Musterklassen für vertikale Aufsummierung. Für die horizontale Summierung verläuft die Musterklassenbildung vergleichbar durch Summierung über die Zeilen.

Verhalten des symbolisierten Signals.

### 7.1.7 Ergebnisse der Musterdetektion

Die Musterklassen zeigen die Anfallseinsätze als auch post-iktale Bereiche im Signal. Für die Musterklasse 3 ist für den Fall der Wechselfrequenz die Musterauftrittshäufigkeit über die Segmente in Bild 7.22 gezeigt. Hier ist auch besonders gut die post-iktale Phase nach einem Anfall zu erkennen, welche in etwa von Segment 150 bis Segment 360 reicht. In Bild 7.23 ist exemplarisch für ein Teilsignal einer EEG-Langzeitmessung gezeigt, welche Musterklassen darin auftreten. Hier sind die Anfallseinsätze auch in einigen der anderen Musterklassen zu erkennen. Die Auswertung mittels der ROC Analyse

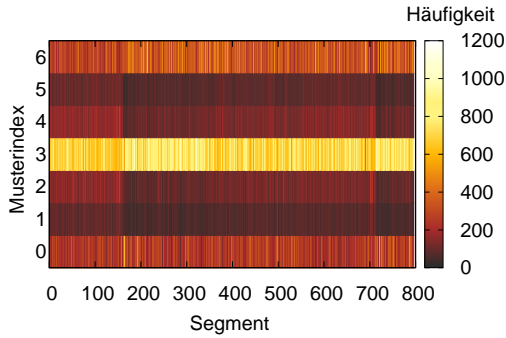


**Bild 7.22:** Musterklasse 3 für die Auszählung der Wechselfrequenz für Datei P. Deutlich ist eine post-iktale Phase zu erkennen und durch Marken (keine Anfallsmarkierungen wie in anderen in dieser Arbeit gezeigten Ergebnissen) bei Segment 150 und Segment 360 hervorgehoben.

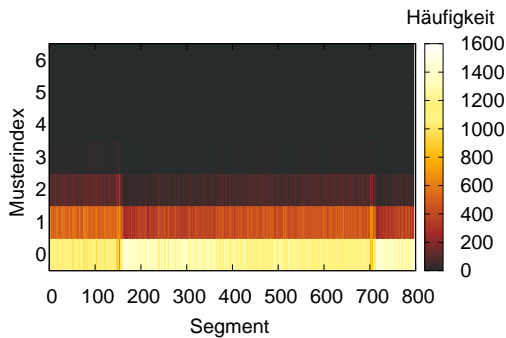
Kanal	TL06	TTL06	TBPR4	TBPL4	TBAR4
AUC-Wert	0.625	0.754	0.616	0.699	0.613
Prädiktions- horizont (Minuten)	360	60	60	80	80

**Tabelle 7.3:** Liste der AUC-Werte für die untersuchten Kanäle der Musterkategorie 0.

zeigte, daß in einigen Fällen hohe AUC-Werte erreicht wurden (vergleiche dazu Bild 7.24), während die Surrogatanalyse nur in wenigen Fällen eine Signifikanz zeigte. Für zwei der untersuchten Kanäle ist das zeitliche Signal über den gesamten Aufnahmezeitraum in Bild 7.25 gezeigt. Es ergaben sich die in Tabelle 7.3 aufgeführten AUC Werte. Angegeben sind hier nur die besten AUC Werte, die sich für



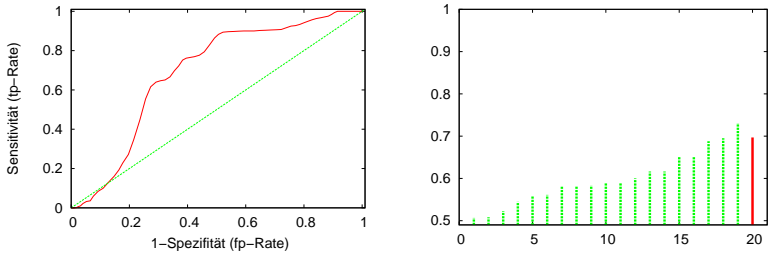
(a) Wechselfrequenz



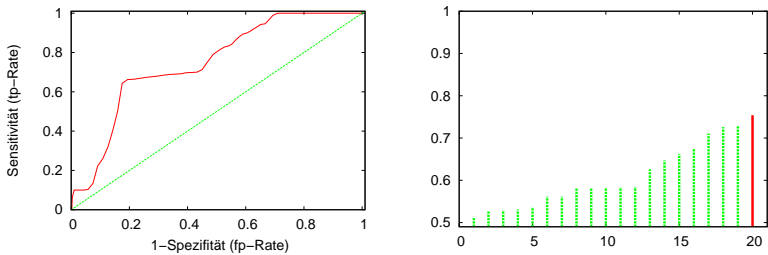
(b) Musterkorrelation

**Bild 7.23:** Musterklassenhäufigkeiten für ein Signal in dem zwei Anfälle auftreten. Diese sind etwa bei den Segmenten 150 und 700 erkennbar. Abgebildet ist die Musterklassenhäufigkeit (a) für die Wechselfrequenz und (b) für die Musterkorrelation.

unterschiedliche Vorhersagehorizonte  $H$  ergaben. Weitere Ergebnisse sind in Anhang A aufgeführt.

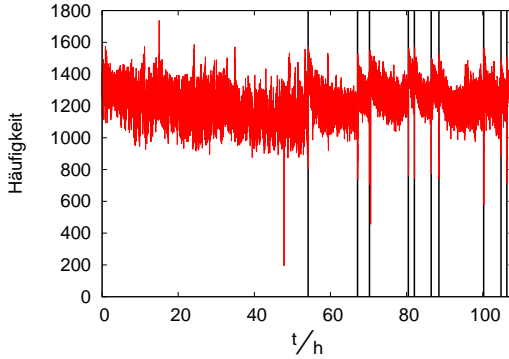


(a)

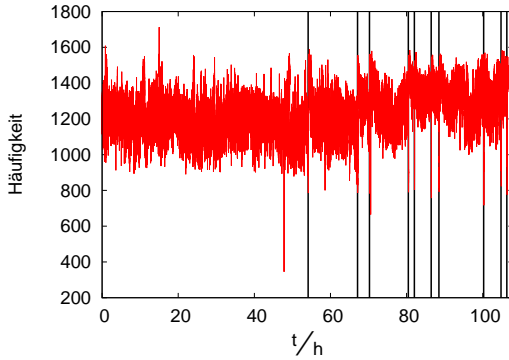


(b)

**Bild 7.24:** Zwei typische Ergebnisse der ROC Analyse für Musterkategorie 0 und Auszählrichtung vertikal (Korrelation der  $1 \times 3$  Muster der Zeilen). Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert des Signals, Linie bei 20 (—), und Ergebnisse der Surrogate, Linien 1 bis 19 (—) des EEG-Signals im Kanal TTL06 (a) und TBPL04 (b).



(a)



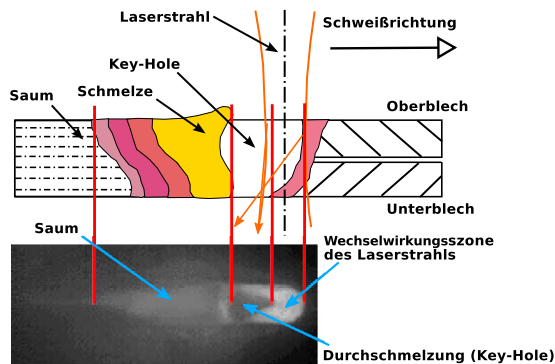
(b)

**Bild 7.25:** Zeitliche Entwicklung der Musterhäufigkeiten in der Musterkategorie 0 für die Kanäle TLL06 (a) und TBPL04 (b). Musterkategoriebildung über die Zeilen der Muster.



## 7.2 Laserschweißen

Die Überwachung von Schweißprozessen z.B. in der Automobilindustrie, ist ein möglicher Einsatz von CNN-basierten Kameras. Dabei wird das Material mit einem Laserstrahl erhitzt, lokal geschmolzen und somit verschweißt. In der Vergangenheit wurde die Qualität einer Schweißnaht erst nach einem abgeschlossenen Schweißvorgang bewertet und konnte durch das Entstehen von schlechten Verschweißungen, zu höheren Kosten führen [86], weshalb eine Steuerung von Laserschweißprozessen bedeutsam ist. Ein automatisches Überwachungs- und Meßsystem für die Güte der entstehenden Schweißnaht kann unter Verwendung von CNN realisiert werden. Durch ein kamerabasiertes System mit Bildraten im Bereich von mehreren Kilohertz bietet sich die Möglichkeit, direkt die Schmelze und deren Dynamik zu beobachten und den Schweißprozess zu steuern [54, 87, 88]. Als Beispiel soll hier auf das Durchschweißen zweier verzinkter Stahlbleche, wie in Bild 7.26 gezeigt, eingegangen werden. Zwei aufein-



**Bild 7.26:** Querschnittsdiagramm einer Durchschweißung nach [54]. Oben Querschnittsdiagramm, unten Bild aus einer Aufnahmesequenz. Korrespondierende Bereiche sind mit Linien verbunden.

ander liegende Bleche werden mit einem Laserstrahl erhitzt, wobei das Material im Fokus des Strahls schmilzt. Ein Transportmechanismus bewegt die zu verschweißenden Teile mit einer definierten Geschwindigkeit voran, so daß letztlich eine Schweißnaht entsteht. Wichtig für die Qualität der Schweißnaht ist die Durchschweißung, welche durch ein sogenanntes Key-Hole in der Wechselwirkungszone, dort wo die Aufschmelzung des Materials erfolgt, erkennbar ist und ein Loch durch beide zu verschweißenden Bleche kennzeichnet. Es kommt dadurch zustande, daß der Laserstrahl im Key-Hole mehrfach reflektiert wird und ein Teil der Laserenergie durch das Werkstück transmittiert wird. Seinen Namen hat dieser Bereich durch seine charakteristische Form, die an ein Schlüsselloch erinnert. Nach [89] teilt sich die Gesamtleistung in die reflektierte Leistung  $P_R$ , die transmittierte Leistung  $P_T$  und die absorbierte Leistung  $P_A$  auf. Hieraus ergibt sich der Einkopplungsgrad zu

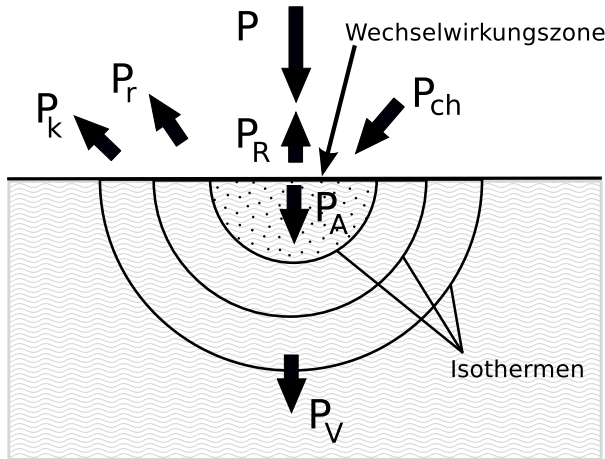
$$\eta_A = \frac{P_A}{P} . \quad (7.3)$$

Die im Werkstück freigesetzte Wärmeleistung  $P_A$  ist zumeist nicht gleich der Prozessleistung  $P_P$  welche für die Verschweißung zur Verfügung steht, sondern ist um die Verlustleistung  $P_V$  vermindert. Wie groß diese Verlustleistung ist, hängt von vielen Faktoren ab. Hierzu zählen unter Anderem das Volumen des Werkstückes, seine Geometrie und seine Materialeigenschaften. Auch thermische Verluste  $P_r \propto T^4$  und Leistungsverluste  $P_k$  durch Luftströmungen tragen zu einer Verringerung der Prozessleistung bei. Allerdings gibt es auch durch chemische Reaktionen, Verbrennungsvorgänge in stark erhitzten Bereichen, zusätzliche Wärmeleistungsbeiträge  $P_{ch}$ . Die genannten Beiträge sind in Bild 7.27 noch einmal zur Verdeutlichung grafisch dargestellt.

Zusammengefaßt ergibt sich hierfür

$$P_A + P_{ch} = P_P + P_k + P_r + P_V . \quad (7.4)$$

Wird  $P_r$  und  $P_k$  als sehr gering und  $P_{ch} = 0$  angenommen so ergibt



**Bild 7.27:** Einkopplung der Laserleistung in der Wechselwirkungszone nach [89].

sich

$$P_P = P_A - P_V \quad (7.5)$$

und daraus der thermische Wirkungsgrad

$$\eta_{th} = \frac{P_P}{P_A} = 1 - \frac{P_V}{P_A} \quad (7.6)$$

Der Prozesswirkungsgrad ist weiterhin durch

$$\eta_P = \frac{P_P}{P} \quad (7.7)$$

gegeben, woraus sich aus 7.3 und 7.6

$$\eta_P = \eta_A \eta_{th} \quad (7.8)$$

ergibt. Durch die Beeinflussung der Größen  $\eta_A$  und  $\eta_{th}$  kann somit die Wirtschaftlichkeit des Prozesses optimiert werden.

Durch das Transportieren der zu verschweißenden Werkstücke entfernt sich der bereits aufgeschmolzene Bereich von der Wechselwirkungszone, kühlt wieder aus und bildet einen Saum. Wenn ein Key-Hole die Durchschweißung erkennen läßt, ist dies ein Indiz dafür, daß die Verschweißung vom Oberblech bis zum Unterblech durchgängig ist und eine feste Verbindung entstanden ist. Daher ist die Durchschweißung ein Gütekriterium für eine gute Schweißnaht, das Fehlen eines Key-Holes bedeutet aber umgekehrt noch nicht eine fehlerhafte Schweißung.

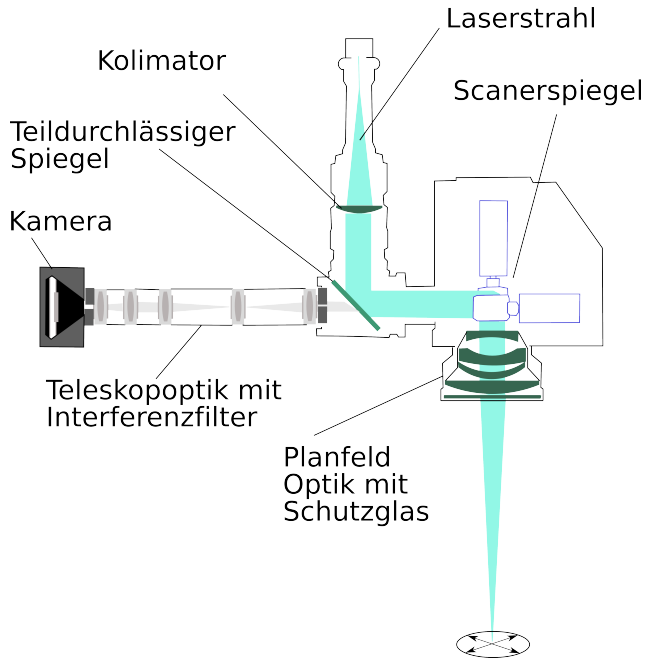
Bei diesem Verfahren können drei wichtige Größen [87] gesteuert werden:

- die Intensität des Laserstrahls,
- die Fokussierung des Laserstrahls,
- die Vorschubgeschwindigkeit des Werkstücks.

Eine optimale Steuerung des Schweißprozesses setzt voraus, daß das Key-Hole sicher detektiert werden kann und bei seinem Fehlen entsprechende Nachregulierungen vorgenommen werden können. Hierfür wurde im Zuge dieser Arbeit ein Algorithmus entwickelt und verbessert, der das Key-Hole in aufeinander folgenden Einzelbildern eines Schweißprozesses detektieren kann. Das Laufzeitverhalten dieses Algorithmus wurde außerdem untersucht.

### 7.2.1 Datenbasis

Die für die Untersuchungen verwendeten Daten bestehen aus Einzelbildern von insgesamt 64 Bildsequenzen, welche am Institut der Forschungsgesellschaft für Strahlwerkzeuge FGSW-Stuttgart mit einer CCD-Hochgeschwindigkeitskamera des Typs Hurricane-40 aufgenommen wurden. Diese wurden zusammen mit Fotos der Ober- und Unterseite des Schweißgutes zur Verfügung gestellt.



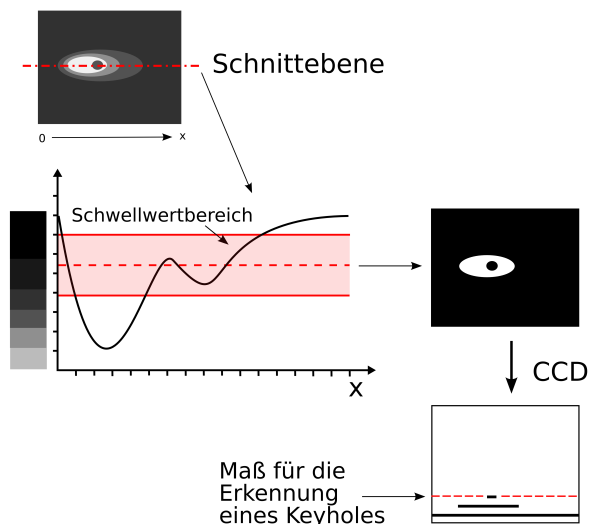
**Bild 7.28:** Koaxiale Aufnahme durch den Strahlengang des Lasers nach [54].

Die Aufnahme erfolgte mittels eines teildurchlässigen Spiegels, wie in Bild 7.28 dargestellt, koaxial zum Strahlengang des Laserstrahls. Um die Bilder auf dem EyeRIS-System verarbeiten zu können, wurden diese auf die Größe von  $128 \times 128$  Bildpunkten gebracht. Bei einer weiteren Messreihe wurde das EyeRIS-System direkt für die Aufnahmen verwendet. Diese Daten stehen ebenfalls als Bildsequenzen zur Verfügung.

## 7.2.2 Algorithmus

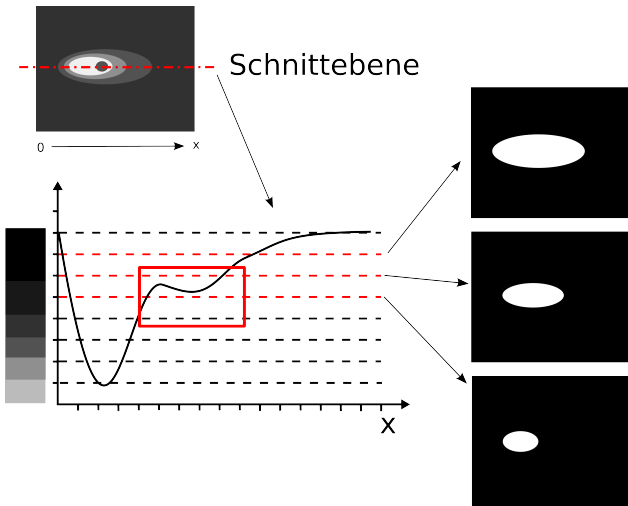
Es wurde ein Algorithmus zur Detektion von Key-Holes, der auf CNN-Hardware lauffähig ist, entwickelt. Der Algorithmus verarbeitet Einzelbilder von Videoaufnahmen der Schweißprozesse. Auf die Einzelbilder wird hierbei eine Schwellwertbildung angewendet. Anschließend werden mittels der CCD-Template die schwarz/weiß-Übergänge (im Folgenden als SW-Übergänge bezeichnet) bestimmt. Die Anzahl der Übergänge kann dann für die Feststellung verwendet werden, ob ein Key-Hole erkannt wurde oder nicht. Der Schwellwert muß hierbei so justiert werden, daß sich ein mögliches Key-Hole deutlich abhebt (siehe Bild 7.29). Eine exakte Justierung ist für eine gute Key-Hole-Detektionsrate von besonderer Bedeutung. Wird der Schwellwert im gezeigten Bereich ungünstig gewählt, so wird die Detektion suboptimal bzw. sogar unmöglich. Der Grauwertbereich von 80 bis 110 stellte sich als besonders geeignet heraus.

Außer dem Schwellwert muß die Anzahl an SW-Übergängen, die man für eine Detektion als signifikant erachtet, festgelegt werden. Dies erfolgt im Falle des in Bild 7.29 gezeigten Beispiels mit einem Schwellwert dadurch, daß für ein Fehlen des Key-Holes ein SW-Übergang weniger vorhanden ist, als wenn dieses vorhanden wäre. Hieraus ergibt sich für die Detektion mit einem Schwellwert, daß ein Key-Hole dann detektiert wird, wenn mindestens drei SW-Übergänge vorhanden sind. In früheren Untersuchungen [54] zeigte sich, daß die Helligkeitsschwankungen während des Schweißens starken Einfluß auf die Detektionseigenschaften hat und eine exakte Einstellung des Schwellwertes in einem schmalen Bereich notwendig ist. Um eine solche diffizile Justierung zu vereinfachen bzw. zu vermeiden, wurde der Algorithmus auf multiple Schwellwerte erweitert. Um ein Key-Hole sicher detektieren zu können, müssen die Pixel, die das Key-Hole repräsentieren, einen Grauwert besitzen, der sich von seiner Umgebung um mindestens eine Schwellwertstufe unterscheidet. Ist dies nicht gegeben, siehe Bild 7.30, sind also die Grauwerte der das Key-Hole repräsentierenden Pixel und die Grauwerte der Pixel in seiner



**Bild 7.29:** Key-Hole Erkennung mit einem Schwellwert. Der Schwellwert muß in einem für den Laserschweißprozess spezifischen Bereich justiert werden.

Umgebung kleiner als der Unterschied zwischen zwei aufeinanderfolgenden Schwellwerten, dann ist eine Detektion im allgemeinen nicht mehr möglich. Ein Key-Hole kann dann dennoch detektiert werden, wenn der Grauwert des Key-Holes knapp über, die Grauwerte der Pixel in seiner Umgebung knapp unterhalb eines Schwellwertes liegen (siehe Bild 7.31). Der hier vorgestellte, erweiterte und in Bild 7.32 schematisch gezeigte Algorithmus zur Detektion von Durchschweißungen bietet hier durch Anwendung mehrerer Schwellwerte einen Lösungsansatz. Der in Bild 7.33 gezeigte erweiterte Algorithmus kann in sechs Schritte aufgeteilt werden, welche jeweils eine spezielle Aufgabe erfüllt und im folgenden genauer erläutert werden. Der erste Schritt initialisiert die den Algorithmus steuernden Größen. Im zweiten Schritt wird das zu verarbeitende Bild geladen

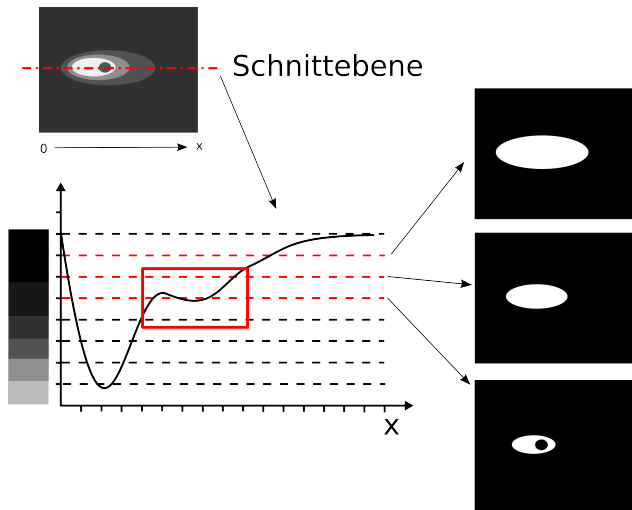


**Bild 7.30:** Das Key-Hole wird nicht detektiert, da der Grauwertunterschied im Bereich des Key-Hole zu gering ist.

oder mittels des optischen Eingangs des ACE16kv2 aufgenommen. Damit Daten eines vorhergehenden Durchlaufs die Verarbeitung des neuen Bildes nicht stören, müssen eventuell verwendete Zwischenspeicher gelöscht werden. Der dritte Schritt führt in einer Schleife Schwellwert-Operationen aus (siehe auch Bild 7.34), deren Ergebnisbilder mittels einer XOR-Operation verknüpft werden (vergleiche Bild 7.32).

Im vierten Schritt werden die SW-Übergänge im XOR verknüpften Ergebnisbild des vorhergehenden Verarbeitungsschrittes ermittelt und als Maß für die Erkennung des Loches verwendet, siehe Bild 7.35. Hierbei ergibt sich die Anzahl an SW-Übergängen, die ein Key-Hole detektieren, aus der Anzahl angewandter Schwellwerte. Im in Bild 7.32 gezeigten Fall wurden drei Schwellwerte angewandt, womit sich drei Linien zusätzlich zur Basislinie ergeben, wenn kein Key-Hole vorhanden ist. Ist ein Key-Hole vorhanden, sollte, wie ge-





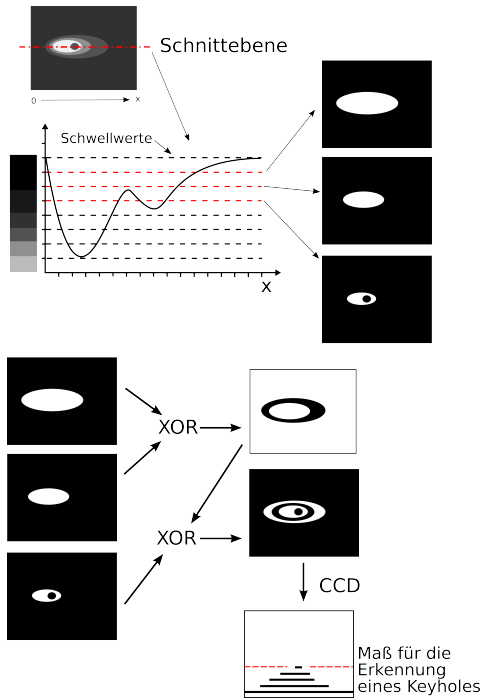
**Bild 7.31:** Das Key-Hole wird detektiert, obwohl der Grauwertunterschied im Bereich des Key-Holes zu gering ist.

zeigt, eine zusätzliche schwarze Linie existieren, also insgesamt 5 Linien. Im fünften Schritt werden weitere Nachverarbeitungsschritte angewendet, um weitere Informationen über den Prozeß aus den gewonnenen Daten zu extrahieren; die eigentliche Regelung wird im sechsten Schritt vorgenommen.

Dieser Prozeß wird für jedes einzelne Bild der Aufnahmesequenz wiederholt, welches entweder auf das EyeRIS-System geladen wird, oder über die Optik aufgenommen werden kann.

### 7.2.3 Laufzeitverhalten

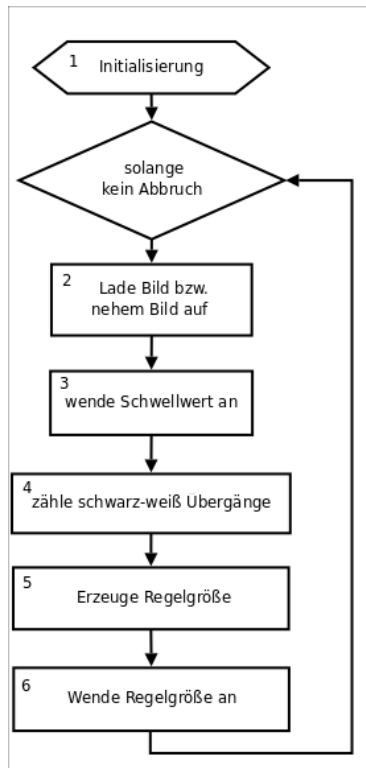
Das Laufzeitverhalten des im vorigen Abschnitt 7.2.2 beschriebenen Algorithmus wird im folgenden untersucht und verschiedene Implementierungen der einzelnen Teilalgorithmen werden vorgestellt.



**Bild 7.32:** Kern der Key-Hole Erkennung: die XOR-Operation. Die Anzahl an SW-Übergängen, die ein Key-Hole detektieren, ergibt sich aus der Anzahl angewandter Schwellwerte. Im gezeigten Fall wurden drei Schwellwerte verwendet, womit sich drei Linien zusätzlich zur Basislinie ergeben, wenn kein Key-Hole vorhanden ist. Ist ein Key-Hole vorhanden, sollte, wie hier gezeigt, eine zusätzliche schwarze Linie existieren, also insgesamt 5 Linien.

Bei den Untersuchungen wurden im Gegensatz zur Erläuterung im vorherigen Abschnitt vier anstelle von drei Schwellwerten verwendet.

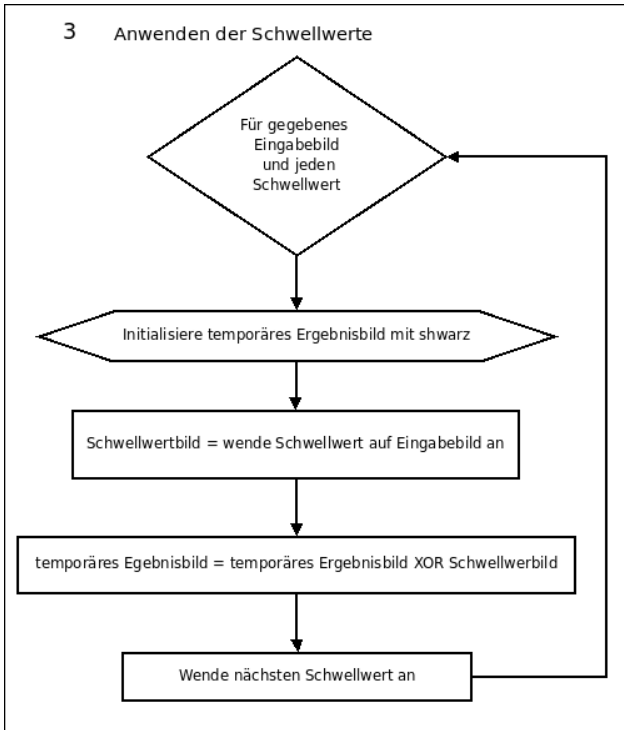
Die Initialisierung ist ein notwendiger Schritt des Algorithmus, der Speicherbereiche reserviert und mit Vorgabewerten belegt und trägt,



**Bild 7.33:** CNN Algorithmus zur Erkennung des Key-Holes. Zur Anwendung des Schwellwertes siehe Bild 7.34. Die einzelnen im Text referenzierten Teilschritte sind durchnummeriert.

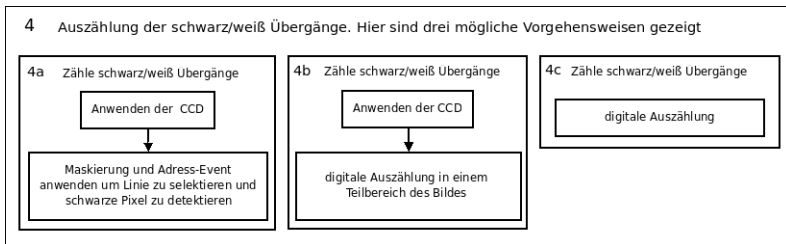
wie der zweite Schritt des Algorithmus, nur mit einer geringen Laufzeit zur Gesamtlaufzeit bei.

Der dritte Schritt des Algorithmus, der die Schwellwerte anwendet, trägt linear mit der Anzahl der anzuwendenden Schwellwerte zur Gesamtlaufzeit bei, weshalb es hier wünschenswert ist, mit so wenig Schwellwerten wie möglich auszukommen. Die Bildgröße wirkt sich dabei je nach Realisierung des CNN unterschiedlich aus. Während



**Bild 7.34:** Anwendung von Schwellwerten und XOR-Operation.

eine schaltungstechnische Realisierung eines CNN, bei der jedes Pixel des Bildes einer Zelle auf dem Chip zugeordnet werden kann, ein Bild parallel verarbeitet und somit die Laufzeit nicht von der Größe des Bildes abhängt. Bei einer Implementierung in Software, sei es eine Simulation eines CNN oder eine digitale Verarbeitung der Bilddaten, hängt dagegen die Laufzeit stark von der Größe des untersuchten Bildes ab, da hier die Anwendung eines Schwellwertes über alle Pixel seriell durchgeführt werden muß. Für ein quadratisches Bild mit Kantenlänge  $n$  ist die Laufzeit proportional  $S \cdot n^2$ , wogegen bei einer Implementierung dieses Teilalgorithmus in Hardware (wie



**Bild 7.35:** Auszählen der schwarz/weiß-Übergänge. Gezeigt sind drei mögliche Implementierungen für diesen Teilalgorithmus. 4a: ausschließlich auf dem analogen Prozessor, 4b: ein hybrider Lösungsansatz, in welchem die CCD auf dem analogen, die Auszählung auf dem digitalen Prozessor des EyeRIS-Systems durchgeführt wird, 4c: rein digital implementierte Auszählung der WS-Übergänge.

z.B. auf einem CNN) die Laufzeit nur linear mit der Anzahl  $S$  der anzuwendenden Schwellwerte wächst.

Der vierte Schritt, in welchem die SW-Übergänge ausgewertet werden, ist sowohl bei einer Realisierung auf dem EyeRIS-System als auch in einer Softwarerealisierung von der Bildgröße abhängig. Dieser Teilalgorithmus kann auf der CCD-Template aufbauen, welche, wie in Abschnitt 2.2.4 beschrieben, in die Kategorie der propagierenden Templates fällt und für das EyeRIS-System in Abschnitt 3.1 genauer erläutert wird. Dies bedeutet für die Ausführung auf der für die vorliegenden Untersuchungen verwendeten Hardwarerealisierung des CNN, daß die Template iterativ mindestens so oft angewendet werden muß, wie das Bild in der betrachteten Richtung Pixel hat, da die einmalige Anwendung der CCD-Template den Bildinhalt nur um ein Pixel, in die durch die Template vorgegebene Richtung, verschiebt. Dies stellt eine Besonderheit des EyeRIS-Systems dar. Ein anderes Vorgehen ist das direkte Auszählen der SW-Übergänge im digitalen Teil des EyeRIS-Systems. Beide Methoden werden in den folgenden Abschnitten genauer untersucht, da sie die Laufzeit des

Algorithmus besonders beeinflussen.

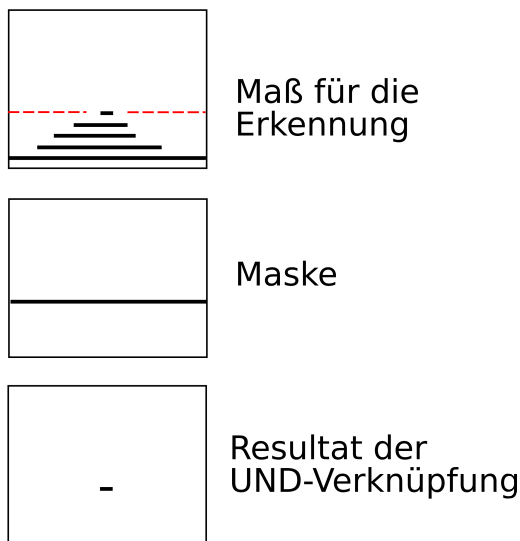
Im fünften Schritt des Algorithmus werden weitere Verarbeitungen durchgeführt, um Größen abzuleiten, die der Prozesssteuerung des Schweißens dienlich sein können. Hier bietet sich z.B. die Auswertung sowohl der vertikalen als auch der horizontalen Anzahl an SW-Übergängen an. Eine Erweiterung des Algorithmus zur Berechnung des Flächeninhaltes der schwarzen Pixel bei unterschiedlichen Schwellwerten kann weiterhin für die Ermittlung der Helligkeitsverteilung im Bild verwendet werden. Je nach Komplexität dieser Algorithmen muß man die Laufzeit getrennt ermitteln und zur Gesamtlaufzeit addieren.

Im sechsten Schritt des Algorithmus können die gefundenen Werte für die Steuerung des Schweißprozesses in Regelgrößen umgesetzt und damit der Schweißprozess an den momentanen Zustand angepaßt werden. Zu regelnde Größen können hierbei [87] die Strahlintensität, der Fokus des Strahls und die Vorschubgeschwindigkeit des Werkstückes sein. Die Anzahl der zu regulierenden Größen trägt näherungsweise linear zur Gesamtlaufzeit bei.

### **Anwendung der CCD-Template und der AdressEvent-Funktion**

Der Schritt 4 des vorgestellten Algorithmus, der das Auszählen der SW-Übergänge durchführt, soll in diesem Abschnitt genauer betrachtet werden. Die CCD-Template ist eine propagierende Template, weshalb diese, wie in Abschnitt 3.1 erläutert, auf dem EyeRIS-System iterativ angewendet werden muß. Daher ist es für ihre Verwendung auf dem EyeRIS-System erforderlich, diese in einer Schleife auszuführen. Da die Bilder auf dem EyeRIS-System bis zu  $128 \times 128$  Pixel groß sind, ist die CCD-Template für horizontale bzw. vertikale Richtung jeweils 128 mal auszuführen, um sicher sein zu können, ein Ergebnis zu erhalten, welches die Gesamtausdehnung des Bildes mit einbezieht. Des Weiteren ist es anschließend nötig, die Anzahl der

schwarzen oder weißen Linien – am unteren Bildrand bei vertikaler CCD nach unten (am oberen Bildrand nach oben) bzw. am linken Bildrand bei horizontaler CCD nach links (am rechten Bildrand nach rechts) – zu zählen. Die Detektion eines Key-Holes kann dann an dieser Stelle durch eine logische UND-Operation mit einer Maske durchgeführt werden, die z.B. die fünfte schwarze Linie markiert, wie in Bild 7.36 gezeigt. Die Anzahl an SW-Übergängen, die ein Key-Hole detektieren, und somit die Maske, die angewendet werden muß, ergibt sich aus der Anzahl verwendeter Schwellwerte. Im in Bild 7.36 gezeigten Fall wurden drei Schwellwerte angewandt, womit sich drei Linien zusätzlich zur Basislinie ergeben, wenn kein Key-Hole vorhanden ist. Ist ein Key-Hole vorhanden, sollte, wie gezeigt, eine zusätzliche schwarze Linie vorhanden sein, also insgesamt 5 Linien. Eine auf



**Bild 7.36:** Anwendung einer Maske zur Detektion von Key-Holes.

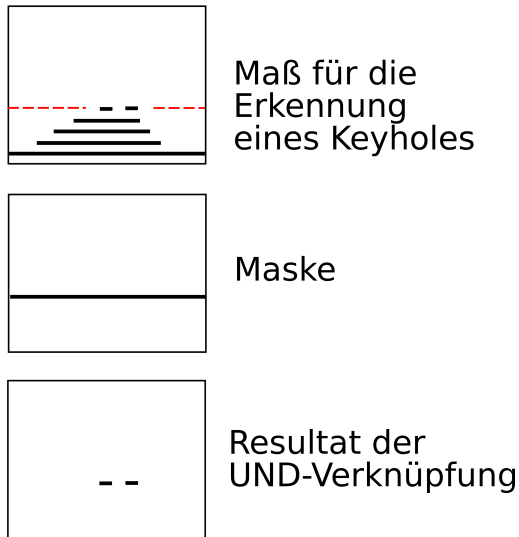
dem EyeRIS-System verfügbare AdressEvent-Funktion kann dann feststellen, ob sich in der fünften Zeile tatsächlich schwarze Pixel be-

finden. Für das EyeRIS-System 1.1 ist dazu erst eine Invertierung des Bildes notwendig, da die AdressEvent-Funktion nur weiße Pixel zählt. Im digitalen Teil des EyeRIS-Systems muß weiterhin die Anzahl der tatsächlich gelesenen Pixel ermittelt werden und durch eine weitere Funktion die Position der durch die AdressEvent-Funktion ermittelten Pixel bestimmt werden, da diese Funktion Pixelkoordinaten im Bild liefert und nicht das Pixel selber. Sollte der Schwellwert an mehreren Stellen überschritten werden, wie in Bild 7.37 illustriert, dann ist es nötig festzustellen, ob eine ausreichende Anzahl an Pixeln als zusammenhängende Linie betrachtet werden kann, um ein Key-Hole als sicher detektiert anzunehmen. In den Untersuchungen stellte sich heraus, daß drei aufeinander folgende schwarze Pixel für eine Detektion ausreichend sind. Um die Detektionsrate zu erhöhen, ergab sich bei Durchsicht der Einzelbilder, daß zusätzlich auch Linien mit mehr als drei schwarzen Pixeln mit vereinzelt Unterbrechungen von maximal zwei weißen Pixeln mitberücksichtigt werden können. Es zeigte sich, daß auf der vorhandenen EyeRIS 1.1-Plattform die CCD-Template den Großteil der Gesamtlaufzeit, wie auch in Tabelle 7.4 ersichtlich, ausmacht. Eine schaltungstechnische Realisierung, bei der das Laufzeitverhalten einer propagierenden Template nicht von der Größe des Bildes abhängt, kann das Laufzeitverhalten dieses Teilalgorithmus beträchtlich verbessern.

### **Auszählung der Übergänge auf dem digitalen Prozessor**

Da Länge und Breite des Key-Holes als ein Merkmal für die Key-Hole Detektion verwendet werden kann, ist es wünschenswert, diese Größen ohne wesentliche zusätzliche Laufzeit zu bestimmen. Hierbei kann im vierten Schritt des vorgestellten Algorithmus anstelle der CCD-Template und anschließender Anwendung einer Maske und der AdressEvent-Funktion eine direkte Zählung der SW-Übergänge im digitalen Teil des EyeRIS-Systems vorgenommen werden. Dies bietet den Vorteil, daß die Länge bzw. Breite eines Be-





**Bild 7.37:** Lücken, die nach der Masken-Operation auftreten können.

reichs gleichzeitig schon während des Auszählens der schwarz/weiß-Übergänge festgestellt werden kann und dies ohne maßgebliche Beeinflussung der Laufzeit mit in die Entscheidung, ob ein Key-Hole detektiert wird oder nicht, einbezogen werden kann. Ein Auszählen muß allerdings im Allgemeinen immer über alle Pixel einer der betrachteten Richtungen des Bildes durchgeführt werden, da nur so sichergestellt werden kann, daß kein SW-Übergang unberücksichtigt bleibt.

Es wurden vier Auszählungs-Algorithmen implementiert und sowohl auf einem handelsüblichen PC als auch auf dem NIOS II Soft-Core-Prozessor auf ihr Laufzeitverhalten untersucht. Sie verwenden zur Detektion von SW-Übergängen die folgenden unterschiedlichen Kriterien:

- es werden die Operatoren  $<$  (kleiner) und  $>$  (größer) zwischen zwei benachbarten Pixeln verwendet, um einen Übergang zu detektieren,
- es wird mittels bitweiser UND-Operation der binären Darstellung benachbarter Pixelwerte und anschließender Auswertung des MSB (Most Significant Bit) unter Zuhilfenahme einer bit-Maske auf einen SW-Übergang geprüft,
- es werden mittels bitweiser UND-Operation die Pixelwerte zweier benachbarter Pixel und anschließender bitweiser Verschiebeoperation um 7 Stellen nach rechts, anstelle der Anwendung einer bit-Maske, eine logische Repräsentation erzeugt, die anzeigt, ob ein SW-Übergang vorliegt. Durch die Verschiebung ist das vorherige MSB nun im LSB (Least Significant Bit) enthalten,
- es werden vier Pixel in einem 32 bit Wort zusammengefaßt und mittels bitweiser UND-Operation von vier benachbarten Pixeln gleichzeitig verglichen. Durch anschließende viermalige Ausführung der bitweisen Verschiebeoperation um 7-bit für das erste Byte und 8-bit für die drei folgenden Bytes des 32 bit Wortes ergeben sich vier logische, boolesche Werte im LSB nach der jeweiligen Verschiebung, je einer für jedes Pixel im 32bit Wort, welche einen schwarz/weiß-Wechsel anzeigen.

Der Algorithmus, welcher mit den Vergleichsoperationen  $<$  und  $>$  arbeitet, wurde aufgrund seiner Einfachheit als Basis zum Vergleich der Algorithmen herangezogen. Zeitmessungen der Algorithmen ergaben die in den Tabellen 7.5 und 7.6 auf Seite 134 zusammengestellten Zeiten. Es ist klar zu erkennen, daß der Algorithmus mit logischem Vergleich des MSB auf beiden Systemen jeweils die Implementierung mit der schnellsten Laufzeit ist.

Teilalgorithmus	Laufzeit (ms)	Bilder/s	Bemerkung
Grauwert Schwellwert	1.049 bis 1.1	909	Bild wurde geladen, 4 Schwellwerte angewendet
Grauwert Schwellwert	0.594 bis 0.6	1666	Bild wurde mit Optical aquisition aufgenommen, 4 Schwellwerte angewendet
Grauwert Schwellwert und Nachbearbeitung	0.771 bis 0.774	1291	Bild wurde mit Optical aquisition aufgenommen, 4 Schwellwerte angewendet und nachbearbeitet
Nachbearbeitung einzeln	0.494 bis 0.496	2016	Beseitigung einzelner Pixel
CCD	4.398 bis 4.4	227	Anwendung der CCD-Template
Algorithmus komplett	4.868 bis 4.9	204	Bild wurde aufgenommen und 4 Schwellwerte sowie CCD angewendet

**Tabelle 7.4:** Laufzeitmessung für einzelne Teilalgorithmen.

Algorithmus	Laufzeit (ms)	Bemerkung
Referenz-Algorithmus	4.504	Vergleiche mit > bzw. < Operator
Logische Operationen	3.176	logische Operationen auf LSB
8 bit-Algorithmus auf Bild	8.235	bitweise UND- mit anschließender Verschiebeoperation um 7 bit nach rechts
32 bit-Algorithmus auf Bild	6.038	bitweise UND-Operation mit anschließender viermaliger Verschiebeoperation um 7 bit nach rechts

**Tabelle 7.5:** Zeitmessung für die digitalen Zählalgorithmen auf dem NIOS II Soft-Core-Prozessor.

Algorithmus	Laufzeit (ms)	Bemerkung
Referenz-Algorithmus	0.338 - 0.379	Vergleiche mit > bzw. < Operator
Logische Operationen	0.266- 0.289	logische Operationen auf LSB
8 bit-Algorithmus auf Bild	1.233 - 1.241	bitweise UND- mit anschließender Verschiebeoperation um 7 bit nach rechts
32 bit-Algorithmus auf Bild	0.678 - 0.690	bitweise UND-Operation mit anschließender viermaliger Verschiebeoperation um 7 bit nach rechts

**Tabelle 7.6:** Zeitmessung für die digitalen Zählalgorithmen auf einem PC (Pentium IV, 1.8 GHz). Im Gegensatz zum Programmlauf auf dem NIOS Prozessor des EyRIS Systems laufen auf einem PC auch andere Programme, weshalb hier Zeitspannen der Laufzeiten angegeben werden.

## Kombination von CCD und Auszählung

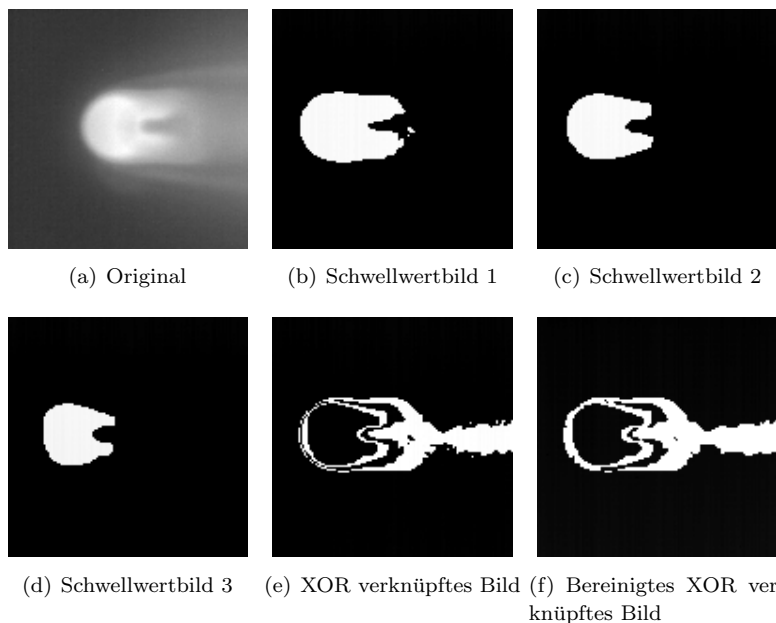
Eine weitere mögliche Implementierung dieses Teilalgorithmus ist die Kombination aus CCD und Auszählalgorithmus, welche zu einem dynamischen Auszählen führt. Hierbei wird wie zuvor die CCD angewandt, dann aber anstelle der AdressEvent-Funktion das Bild im digitalen Teil des EyeRIS-Systems ausgezählt. Hierbei wird die Auszählung gestoppt, wenn keine SW-Übergänge in der betrachteten Richtung im Ergebnisbild der CCD mehr vorhanden sind, oder wenn ein Abbruchkriterium – z.B. eine vorgegebene Anzahl an SW-Übergängen, die als sicheres Maß für die Detektion eines Key-Holes festgelegt wurden – erfüllt ist.

### 7.2.4 Ergebnisse

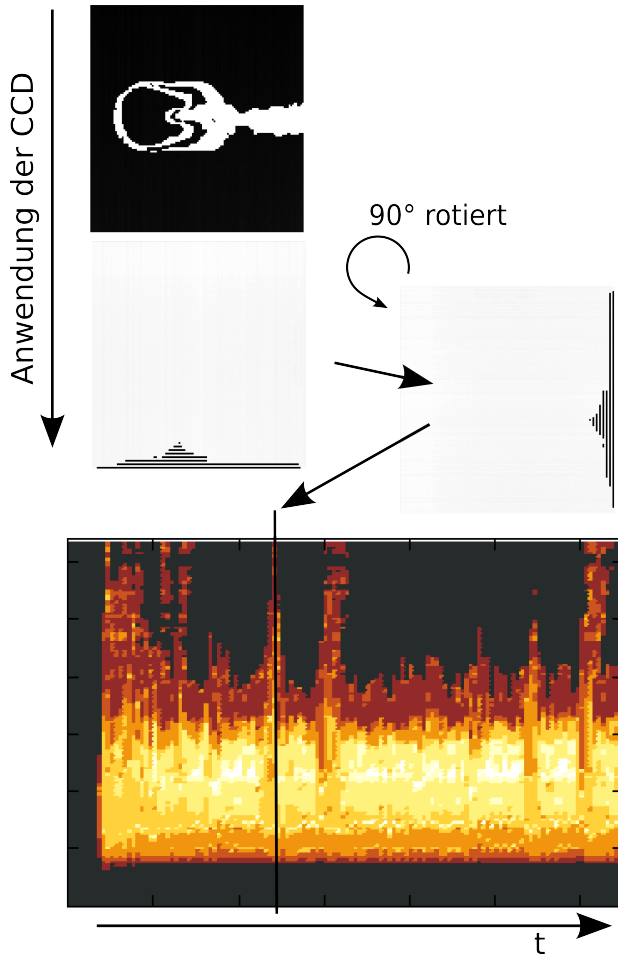
Die Einzelbilder der Bildsequenzen wurden mittels des Key-Hole-Detektionsalgorithmus bearbeitet. In Bild 7.38 sind exemplarisch die Einzelschritte der Key-Hole Detektion an einem Einzelbild gezeigt. Um die zeitliche Entwicklung besser darstellen zu können, wird die Anzahl an SW-Übergängen in den jeweiligen Bildern farblich kodiert, um 90° gedreht und in zeitlicher Ordnung zusammengefaßt (siehe Bild 7.39).

Im Folgenden werden drei typische Ergebnisse des Key-Hole-Detektionsalgorithmus diskutiert. In allen Bildern ist die Ober- bzw. die Unterseite des verschweißten Bleches gezeigt. Mittig dazu ist jeweils eine Zusammenfassung der Ergebnisse des Key-Hole-Detektionsalgorithmus dargestellt, die der Schweißnaht entsprechend ausgerichtet ist. Helle Bereiche zeigen hierbei die Güte der Detektion eines Key-Hole an. Je heller, desto präziser konnte ein Key-Hole erkannt werden.

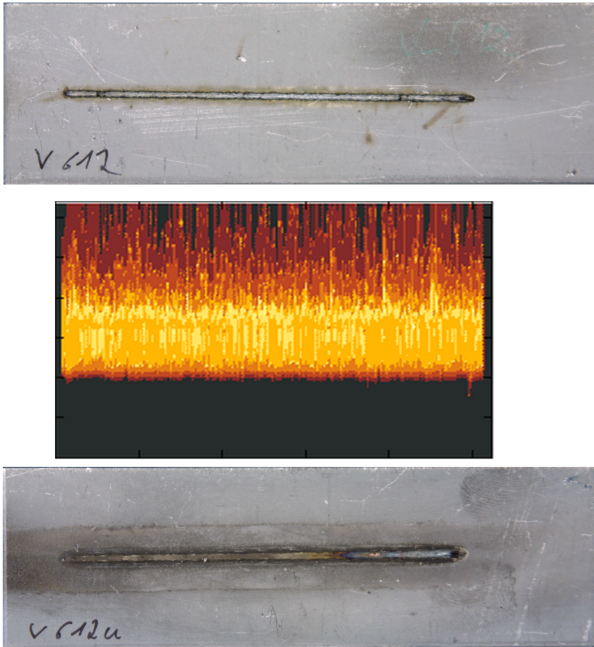
Bild 7.40 zeigt eine Schweißung für ein gutes Schweißergebnis. Deutlich ist hierbei im mittleren Teil ein sehr heller Streifen, der für die Detektion eines Key-Holes steht, zu erkennen.



**Bild 7.38:** Ergebnis der Schwellwertbildung (a-d), XOR-Verknüpfung (e), Small-Pixel-Removal (f), sowie Anwendung der CCD-Template nach rechts (g) und nach unten (h) auf dem ACE16kv2 des EyeRIS-Systems berechnet.



**Bild 7.39:** Die Ergebnisse der Einzelbilder werden um  $90^\circ$  gedreht und in zeitlicher Aufeinanderfolge farblich kodiert zusammengefaßt. Je heller, desto eindeutiger ist ein Key-Hole im Ausgangsbild detektiert worden.

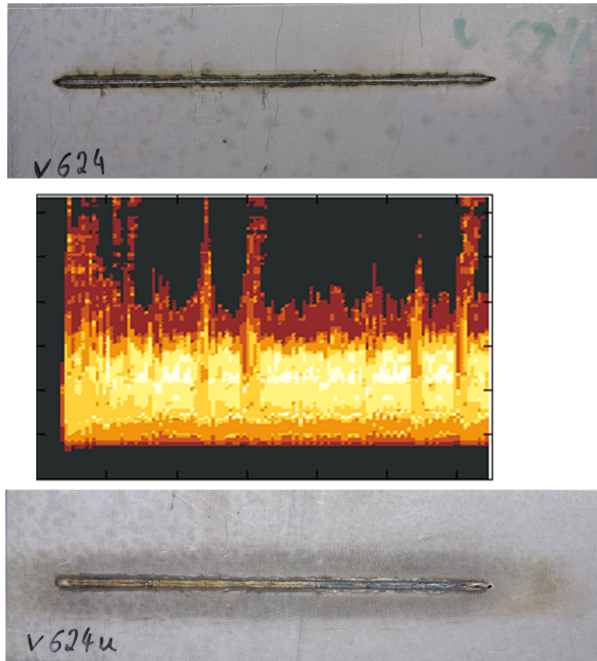


**Bild 7.40:** Ergebnis einer guten Verschweißung. Gezeigt ist Ober- und Unterseite der verschweißten Bleche sowie in der Mitte die Auswertung der Einzelbilder der Aufnahme des Schweißvorganges.

Bild 7.41 zeigt eine Schweißung, bei der häufig Helligkeitsschwankungen in den Einzelbildern der Aufnahmesequenz vorhanden waren. Trotz allem ist hier eine deutliche Detektion eines Key-Holes vorhanden.

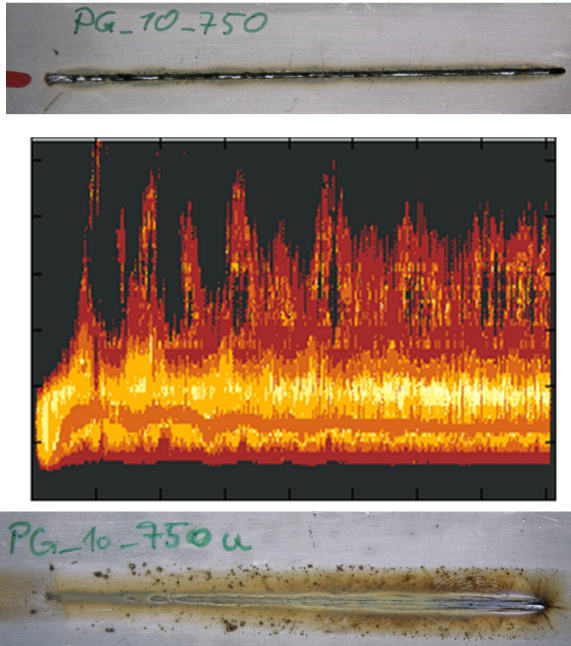
In Bild 7.42 ist eine Schweißung wiedergegeben, bei der die Durchschweißung nicht überall vorhanden ist. Deutlich ist hier im Bild der Key-Hole-Detektion zu erkennen, daß eine Durchschweißung nicht überall sauber erkannt werden konnte. Hierbei stellte die schwankende Gesamthelligkeit von Einzelbildern, wenn z.B. während des Schweißens glühende Metallschmelze wegspritzte, die größte Fehler-





**Bild 7.41:** Schweißprozess mit Fluktuation der Helligkeit. Gezeigt sind Ober- und Unterseite der verschweißten Bleche, sowie die Auswertung der Einzelbilder in der Mitte. In der Mitte ist der zeitliche Verlauf der Key-Hole Detektion gezeigt. Während des Schweißprozesses kam es zu deutlichen Schwankungen in der Helligkeit der Aufnahmen.

quelle dar. Für die Auswertung der Ergebnisse wurden aus den 64 Sequenzen, mit insgesamt 12376 Bildern, 798 Bilder als Stichprobe auf optische Erkennbarkeit von Durchschweißungen durch das menschliche Auge untersucht und mit den Detektionen durch den Algorithmus verglichen. Hierbei wurden, wie im Kapitel 6 beschrieben, True Positives und True Negatives als auch die False Positives und False Negatives ermittelt. Die Auswertung der 798 Bilder ergab 73



**Bild 7.42:** Diese Sequenz zeigt einen Schweißprozess, der suboptimal verlaufen ist. An der Ober- und Unterseite der Bleche ist deutlich zu erkennen, daß die Verschweißung nicht durchgängig ist, was man auch in den Ergebnissen des Key-Hole-Detektionsalgorithmus deutlich erkennen kann.

Fehldetektionen. Die Nullhypothese, daß der Algorithmus eine größere Fehldetektionsrate als 10% hat, konnte mit einem Signifikanzlevel von 5%,  $\alpha$ -Fehler, aufgrund des Umfangs der Meßreihe, zurückgewiesen werden. Die Ermittlung des Fehlers 2. Art, wie in Kapitel 6 beschrieben, ergab weiterhin eine 96%ige Wahrscheinlichkeit, für die Richtigkeit der Zurückweisung der Nullhypothese.

## 8 Zusammenfassung

In der vorliegenden Arbeit wurden CNN-Algorithmen, sowohl in der Simulation auf einem PC als auch insbesondere in einer schaltungstechnischen Implementierung von CNN, auf ihre Eignung zur algorithmischen Vorhersage von epileptischen Anfällen sowie zur Überwachung von Laserschweißprozessen untersucht.

Zunächst wird ein Überblick über die unterschiedlichen CNN-Typen gegeben und die Komponenten, die ein solches Netzwerk ausmachen, diskutiert. Besonders wichtig für die durchgeführten Untersuchungen sind hier Discrete-Time-CNN in Kombination mit Time-Delay-CNN. In einem weiteren Abschnitt wird das verwendete EyeRIS-System, mit den sich bietenden Möglichkeiten, vorgestellt. Es zeichnet sich dadurch aus, daß es einen analogen CNN-Chip, den ACE16kv2, und eine digitale CPU miteinander kombiniert, die beide programmiert werden können und somit die Entwicklung von Programmen ermöglicht, die sowohl analoge als auch digitale Teilalgorithmen enthalten. Die sich hieraus ergebenden Möglichkeiten der Datenverarbeitung mit CNN werden diskutiert und ihre Grenzen aufgezeigt. Anschließend werden die zur Auswertung benötigten Methoden vorgestellt. Diese dienen der Untersuchung und Quantifizierung der Aussagekraft der erhaltenen Ergebnisse.

Es folgt ein Kapitel, in dem die Eigenschaften eines neu entwickelten, evolutionär motivierten Optimierungsverfahrens (SSAPE) aufgezeigt werden, welches speziell auf die Optimierung von CNN-Templates auf der analogen Hardware ausgerichtet ist. Die vielen CNN-Anwendungen inhärenten Symmetrien werden durch das Verfahren

automatisch mit berücksichtigt, da sie in die Optimierung mit einbezogen werden können. Das im Zuge dieser Arbeit entwickelte und implementierte SSAPE-Verfahren wurde der Simulationsumgebung FORCE++ hinzugefügt. Durch Untersuchung und Vergleich mit Ergebnissen die mit dem Differential Evolution (DE), einem ebenfalls evolutionär inspirierten Optimierungsalgorithmus, erlangt wurden, kann gezeigt werden, daß es für die hier untersuchten Vorhersagealgorithmen geeignet ist. Die Anwendung von CNN-Algorithmen werden in zwei Anwendungsbereichen demonstriert:

Im medizinischen Bereich wird die Möglichkeit zur Vorhersage von epileptischen Anfällen mit CNN untersucht. Zielsetzung ist es zu zeigen, daß Vorhersagealgorithmen auf CNN-Basis auch auf hardwarebasierten CNN Implementierungen lauffähig sind, was für zukünftige medizinische Anwendungen in tragbaren Geräten ein wesentlicher Aspekt ist. Die Genauigkeitsanforderungen an derartige Schaltungen wird untersucht, um die Anforderungen an CNN-Hardware für solche Geräte besser bestimmen zu können. Hierzu werden zwei Verfahrensansätze (siehe Abschnitt 7.1) auf dem EyeRIS-System implementiert. Eines basiert auf der Optimierung von Prädiktionsparametern und wird auf dem EyeRIS-System mit zwei evolutionär inspirierten Verfahren, SSAPE und DE, untersucht, das andere basiert auf Mustererkennung. Für das erste Verfahren werden die SSAPE und DE Optimierungsalgorithmen auf dem EyeRIS-System implementiert. Ferner werden die Ergebnisse mit denen aus digitalen Simulationen der zwei evolutionären Verfahren verglichen, die sowohl mit doppelter Gleitkommagenauigkeit als auch mit auf 8 bit reduzierter Genauigkeit durchgeführt werden. Die Resultate werden dann mittels ROC-Analyse unter Verwendung von Surrogaten ausgewertet. Hierbei zeigt sich, daß die Auflösungsgenauigkeit einschließlich der Auflösungsverluste von bis zu 3 bit des analogen CNN-Chips dazu führt, daß eine Vorhersage mit diesem Verfahren keine signifikanten Resultate liefert. Die Ergebnisse der zwei Optimierungsverfahren werden gegenübergestellt und diskutiert. Das zweite Verfahren, welches auf der Symbolisierung von Signalen mit anschließen-

---

der Musterdetektion beruht, wird implementiert und die Musterdetektion auf dem EyeRIS-System untersucht. Es werden Musterklassen eingeführt, welche die Anordnung von Mustern – die Einbettung des zeitlichen Signals in eine räumliche Struktur – mit statistischen Auswertungen der Musterkategorien kombiniert. Die statistische Auswertung wird hierbei auf dem digitalen Prozessor des EyeRIS-Systems durchgeführt, der mit 50 MHz getaktet ist; daraus folgt, daß die statistische Auswertung der Musterdaten durch die Taktrate des digitalen Teils des EyeRIS-Systems beschränkt ist. Es erweist sich auch hier, daß Algorithmen generell an die jeweilige schaltungstechnische Realisierung angepaßt und Hardwarebeschränkungen mitberücksichtigt werden müssen, um aussagekräftige Ergebnisse zu ermöglichen.

Im technischen Bereich wird die Überwachung von Laserschweißvorgängen untersucht. Hierzu wird ein neuartiger, auf CNN basierender, Algorithmus entwickelt, der ein Maß für die Güte einer I-Naht-Überlapp-Schweißstoß Verbindung zweier übereinander liegender verzinkter Stahlplatten detektiert. Hierbei kommen verschiedene, für die Bildverarbeitung geeignete Templates zum Einsatz. Die Ergebnisse belegen, daß das entwickelte Verfahren in der Lage ist, Durchschweißungen mit einer Bildrate von etwas über 200 Bildern pro Sekunde zu detektieren. Hier war vor allem die Anwendung der CCD-Template die Laufzeit bestimmende Operation, da der ACE16kv2 Chip propagierende Templates nicht direkt abbilden kann. Ein auf der Simulation des Chua-Yang CNN Modells vergleichbares Ergebnis ist auf dem Chip-System daher nur durch iterative Anwendung der Template zu erlangen. Dieses Verhalten gilt für jede propagierende Template auf dem EyeRIS-System und bildet keine Einschränkung des Algorithmus, sondern ist eine Eigenheit des zur Verfügung stehenden EyeRIS-Systems. Da für eine derartige Prozessüberwachung eine möglichst hohe Bildrate wünschenswert ist, wird das Laufzeitverhalten genauer untersucht und es werden unterschiedliche Implementierungen verglichen. Hierbei wird auch die rein digitale Implementierung des Algorithmus auf einem PC betrach-

tet. Die funktionsfähige Implementierung auf dem EyeRIS-System ist nach den Ergebnissen dieser Untersuchungen langsamer als eine rein digitalen Implementierung, was aber keineswegs generell für CNN-Hardware zu gelten hat.

Wird eine schnelle Datenverarbeitung mit geringen Anforderungen an die Genauigkeit benötigt, ist die hier verwendete CNN-Realisierung geeignet, sofern man die Datenaufnahmen ebenfalls auf dem EyeRIS-System durchführen kann. Bei Anwendungen, in denen man auf präzisere Ergebnisse angewiesen ist, muß man für den jeweiligen Algorithmus prüfen, ob die erreichbare Genauigkeit ausreichend ist.

Das untersuchte Prädiktionsverfahren könnte von CNN-Schaltungen mit höheren Darstellungsgenauigkeiten profitieren. Das Musterdetektionsverfahren kann bei den momentan verfügbaren Systemen durch schnellere digitale Prozessoren in hybriden Systemen (CNN + CPU) oder durch voll zeitkontinuierliche CNN, die eine Ausführung der PatternMatchingFinder-Template ermöglichen, verbessert werden. Hier ist die Entwicklung von CNN-Hardware und ihrer Genauigkeit als auch die Integration mit herkömmlichen, aber schneller getakteten digitalen Prozessoren abzuwarten.

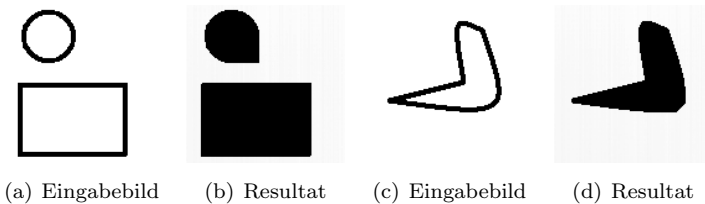
Um die Geschwindigkeit der Bildverarbeitung durch den Algorithmus zur Überwachung und Steuerung eines Schweißprozesses zu erhöhen, bietet es sich an, eine parallelisierte digitale Implementierung für Graphical Processing Units (GPU's) vorzunehmen, die in den heute verfügbaren Grafikkarten programmierbar sind. Untersuchungsergebnisse, die einen Ausblick hierzu geben, sind in Abschnitt A.4 im Anhang gezeigt. Alternativ könnte man einen CCD-Sensor entwickeln, der die Schwellwertbildung und die verwendete XOR-Verknüpfung bereits während des Auslesens der Bilddaten ausführt. Eine CNN-Schaltung bei der auch propagierende CNN-Templates schnell abgearbeitet werden, wäre ein weiterer Lösungsansatz, um die Verarbeitungsgeschwindigkeit des untersuchten Algorithmus zu erhöhen.

# A Anhang mit weiteren Untersuchungsergebnissen

In diesem Anhang sind weitere Ergebnisse zum Optimierungsverfahren und zu den Untersuchungen zur Epilepsievorhersage sowie Untersuchungen des Bildverarbeitungsalgorithmus zur Detektion von Key-Holes bei Laserschweißprozessen aufgeführt.

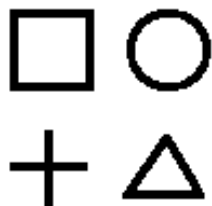
## A.1 Optimierungsverfahren

In diesem Anhang werden weitere Ergebnisse zum Kapitel 5 „Optimierungsverfahren“ gezeigt. In den Bildern A.1 und A.2 sind Ergebnisse der Anwendung der auf dem EyeRIS-System mittels SSAPE optimierten Hole-Filler-Verarbeitung an einzelnen Objekten als auch an mehreren in einem Bild enthaltenen Objekte dargestellt. Die

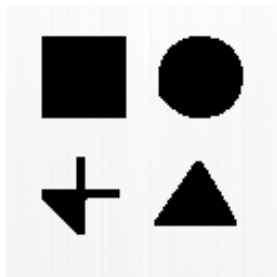


**Bild A.1:** Beispiel für die Anwendung der Hole-Filler-Verarbeitung.

Entwicklung des Fehlers während der Optimierung der Hole-Filler-Verarbeitung auf dem EyeRIS-System ist in A.3 gezeigt.

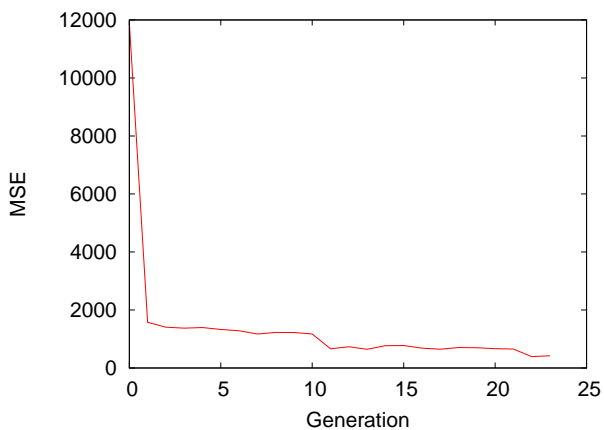


(a) Eingabebild mit mehreren Objekten



(b) Resultat

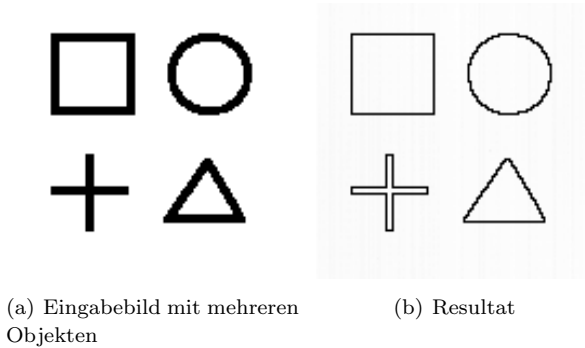
**Bild A.2:** Beispiel für die Anwendung der Hole-Filler-Verarbeitung bei mehreren Objekten auf dem EyeRIS-System.



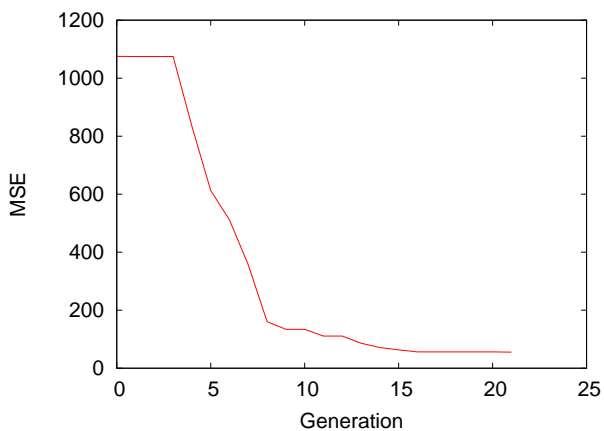
**Bild A.3:** Fehlerentwicklung während der Optimierung bei der Hole-Filler-Verarbeitung für SSAPE bei Optimierung auf mehrere Objekte.



Für die Kantendetektion ergeben sich für eine Optimierung auf dem EyeRIS-System die in A.4 gezeigten Ergebnissbilder und in A.5 gezeigte Entwicklung des Fehlers während der mit SSAPE durchgeführten Optimierung auf dem EyeRIS-Systems.



**Bild A.4:** Beispiel für die Anwendung der Edge-Detection-Verarbeitung bei mehreren Objekten auf dem EyeRIS-System.



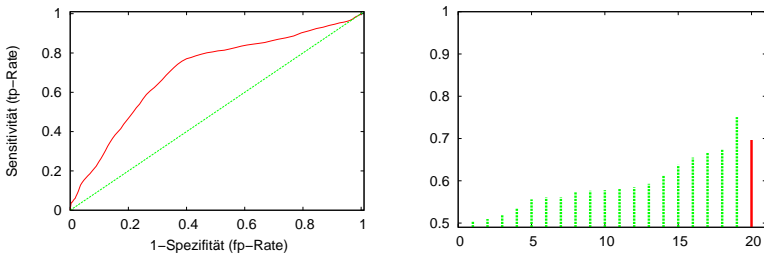
**Bild A.5:** Fehlerentwicklung während der Optimierung bei der Hole-Filler-Verarbeitung für SSAPE bei Optimierung auf mehrere Objekte.

## A.2 Epilepsievorhersage mit Prädiktor

Weitere Ergebnisse zur Vorhersage von epileptischen Anfällen mit Prädiktoren für einen Prädiktionshorizont von 120 Minuten.

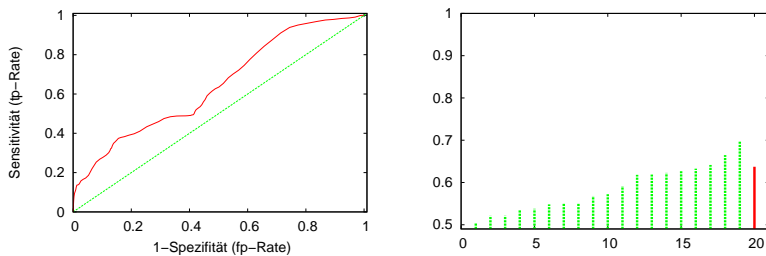
Name bzw. Verfahren	AUC	AUC für bestes Surrogat
SSAPE EyeRIS	0.696	0.750
DE EyeRIS	0.637	0.696
SSAPE 8 bit	0.580	0.640
DE 8 bit	0.581	0.659

**Tabelle A.1:** AUC Werte der verschiedenen Optimierungsverfahren für Kanal TBAR4. Aufgeführt sind die Ergebnisse für die Berechnungen auf dem EyeRIS-System und einer Simulation mit 8 bit Auflösung



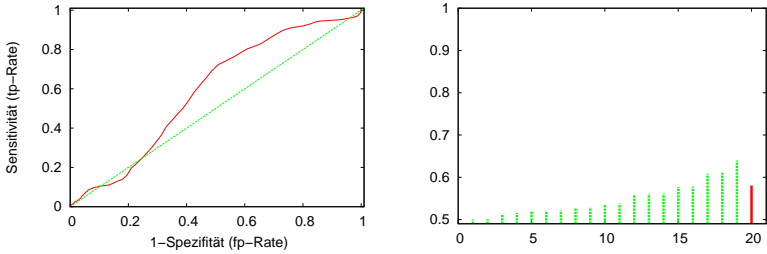
(a)

**Bild A.6:** Ergebnisse für EyeRIS System mit SSAPE für Kanal TBAR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



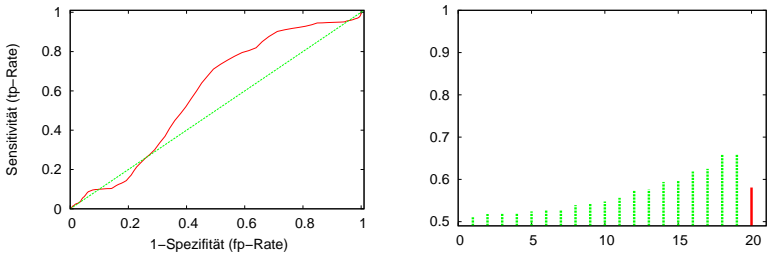
(a)

**Bild A.7:** Ergebnisse für EyeRIS System mit DE für Kanal TBAR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



(a)

**Bild A.8:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit SSAPE für Kanal TBAR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

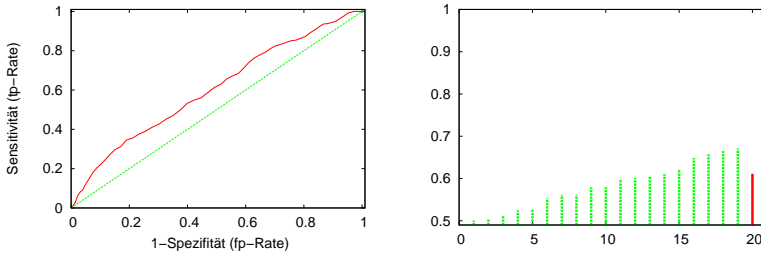


(a)

**Bild A.9:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit SSAPE für Kanal TBAR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

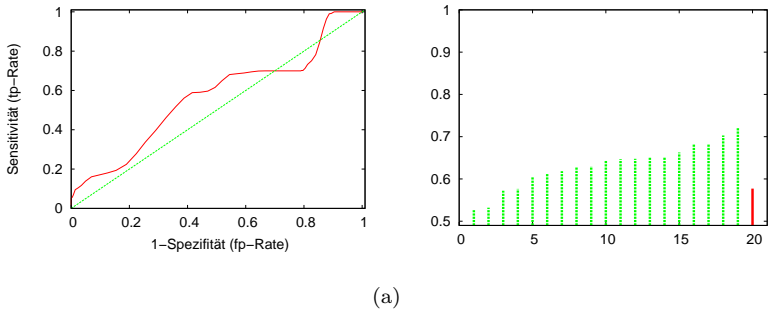
Name bzw. Verfahren	AUC	AUC für bestes Surrogat
SSAPE EyeRIS	0.611	0.671
DE EyeRIS	0.577	0.720
SSAPE 8 bit	0.685	0.687
DE 8 bit	0.681	0.703

**Tabelle A.2:** AUC Werte der verschiedenen Optimierungsverfahren für Kanal TBPL4. Aufgeführt sind die Ergebnisse für die Berechnungen auf dem EyeRIS-System und einer Simulation mit 8 bit Auflösung

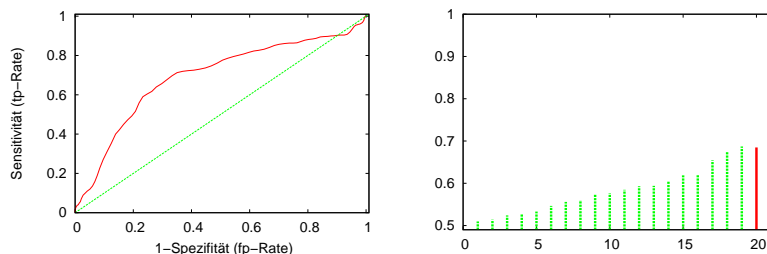


(a)

**Bild A.10:** Ergebnisse für EyeRIS System mit SSAPE für Kanal TBPL4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

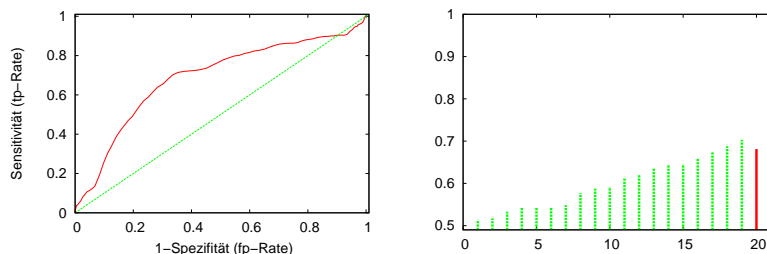


**Bild A.11:** Ergebnisse für EyeRIS System mit DE für Kanal TBPL4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



(a)

**Bild A.12:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit SSAPE für Kanal TBPL4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



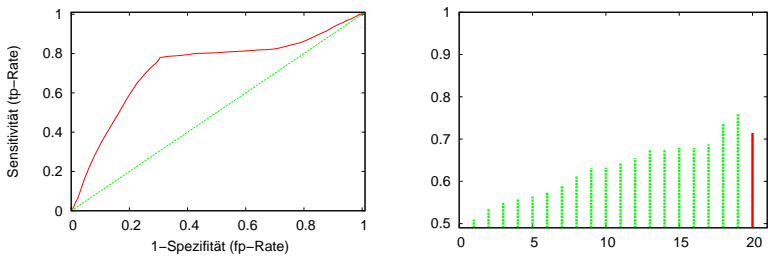
(a)

**Bild A.13:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit DE für Kanal TBPL4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



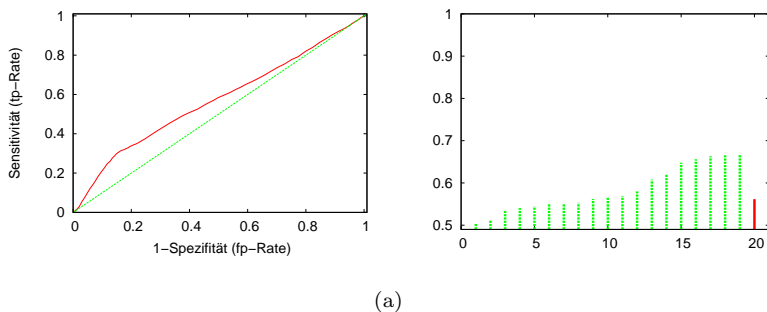
Name bzw. Verfahren	AUC	AUC für bestes Surrogat
SSAPE EyeRIS	0.714	0.761
DE EyeRIS	0.562	0.668
SSAPE 8 bit	0.646	0.698
DE 8 bit	0.649	0.699

**Tabelle A.3:** AUC Werte der verschiedenen Optimierungsverfahren für Kanal TBPR4

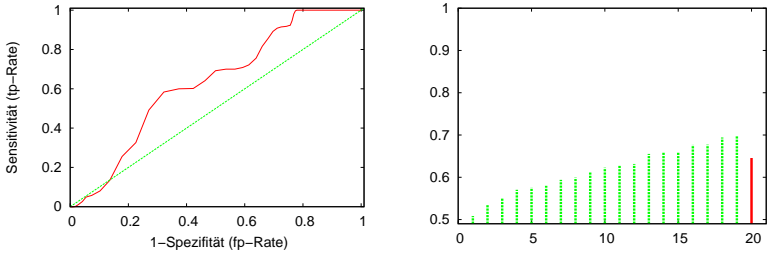


(a)

**Bild A.14:** Ergebnisse für EyeRIS System mit SSAPE für Kanal TBPR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (---).

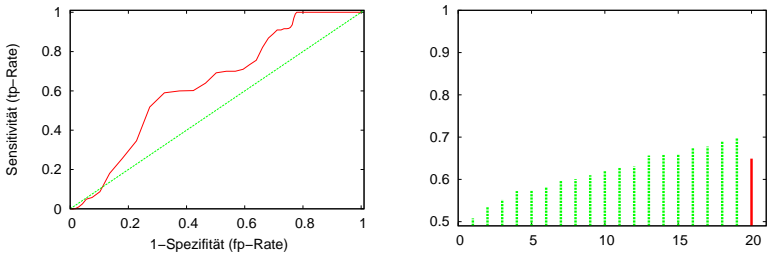


**Bild A.15:** Ergebnisse für EyeRIS System mit DE für Kanal TBPR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



(a)

**Bild A.16:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit SSAPE für Kanal TBPR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

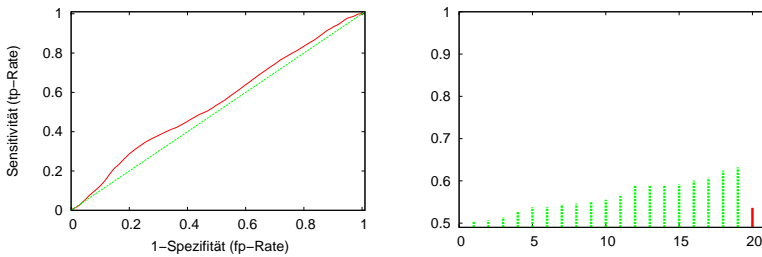


(a)

**Bild A.17:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit DE für Kanal TBPR4. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

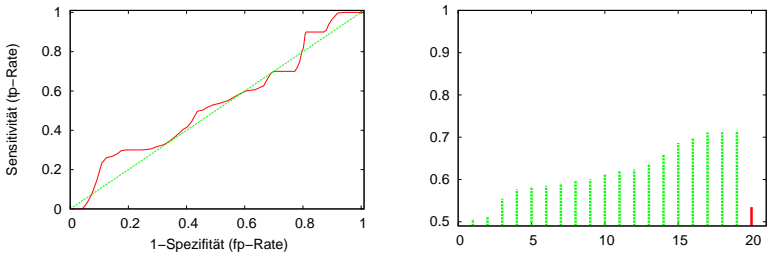
Name bzw. Verfahren	AUC für Kanal TL06	AUC für Surrogate
SSAPE EyeRIS	0.536	0.632
DE EyeRIS	0.535	0.715
SSAPE 8 bit	0.615	0.617
DE 8 bit	0.612	0.618

**Tabelle A.4:** AUC Werte der verschiedenen Optimierungsverfahren für Kanal TLL06



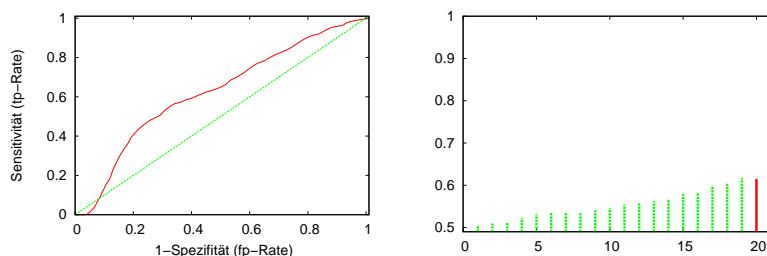
(a)

**Bild A.18:** Ergebnisse für EyeRIS System mit SSAPE für Kanal TLL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



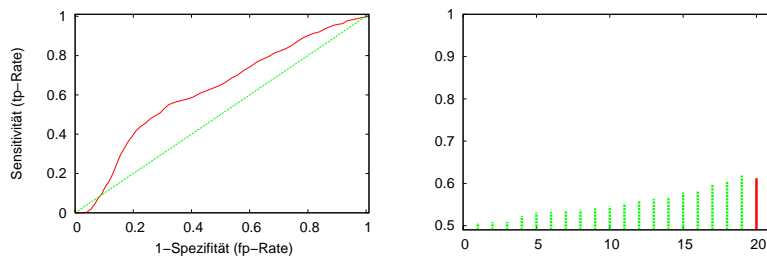
(a)

**Bild A.19:** Ergebnisse für EyeRIS System mit DE für Kanal TLL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



(a)

**Bild A.20:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit SSAPE für Kanal TLL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

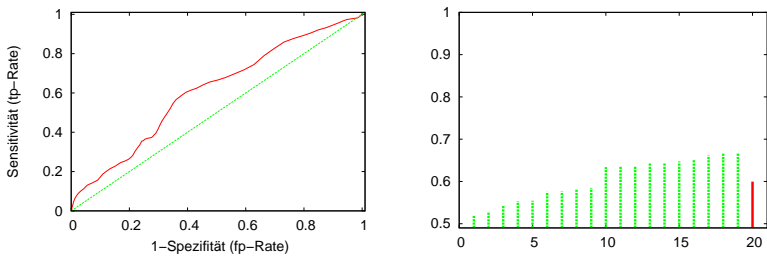


(a)

**Bild A.21:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit DE für Kanal TLL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

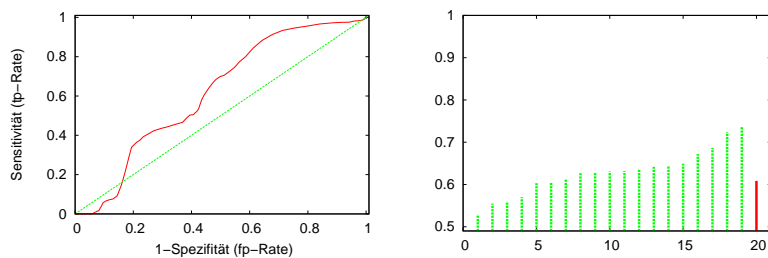
Name bzw. Verfahren	AUC	AUC für bestes Surrogat
SSAPE EyeRIS	0.599	0.668
DE EyeRIS	0.609	0.736
SSAPE 8 bit	0.651	0.667
DE 8 bit	0.655	0.628
SSAPE Gleitkomma	0.651	0.673
DE Gleitkomma	0.645	0.655
BFGS Gleitkomma	0.643	0.665
Simulated Annealing Gleitkomma	0.724	0.616

**Tabelle A.5:** AUC Werte der verschiedenen Optimierungsverfahren für Kanal TL06



(a)

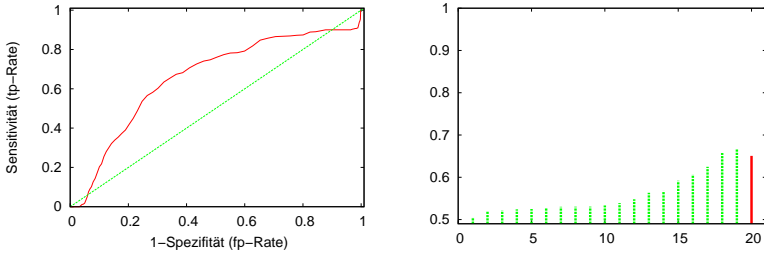
**Bild A.22:** Ergebnisse für EyeRIS System mit SSAPE für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—)



(a)

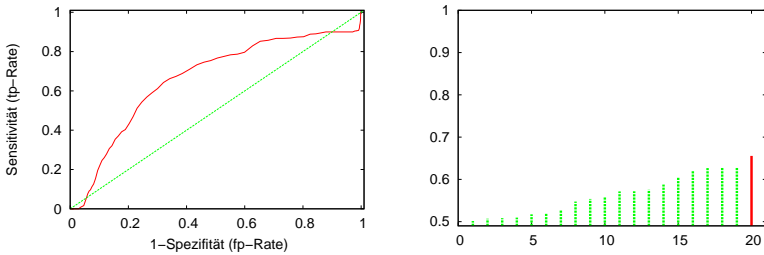
**Bild A.23:** Ergebnisse für EyeRIS System mit DE für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).





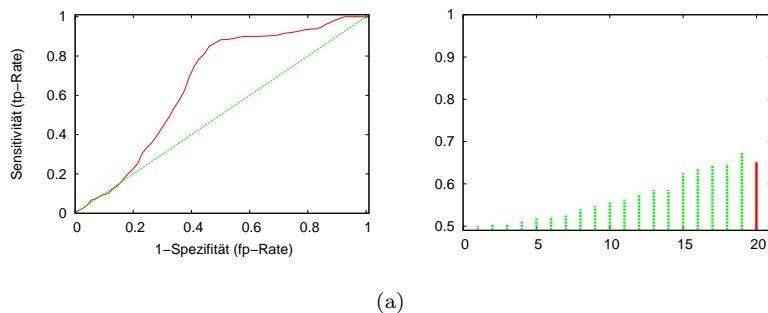
(a)

**Bild A.24:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit SSAPE für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

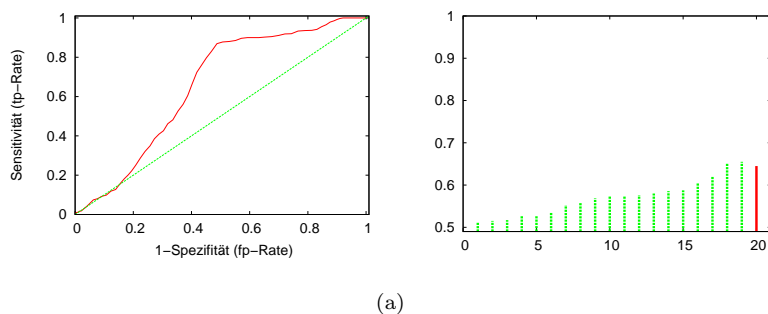


(a)

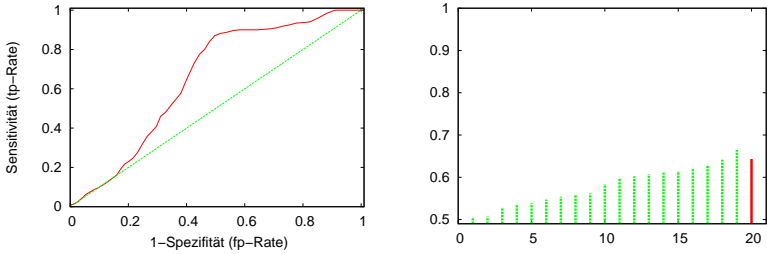
**Bild A.25:** Ergebnisse aus Simulation mit 8 bit Genauigkeit mit DE für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



**Bild A.26:** Ergebnisse aus Simulation mit voller Gleitkommengenauigkeit mit SSAPE für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

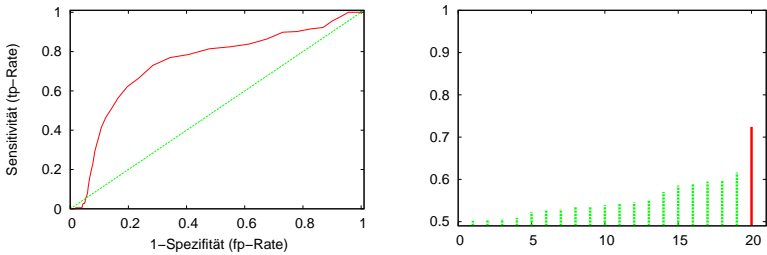


**Bild A.27:** Ergebnisse aus Simulation mit voller Gleitkommengenauigkeit mit DE für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).



(a)

**Bild A.28:** Ergebnisse aus Simulation mit voller Gleitkommengenauigkeit mit BFGS für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

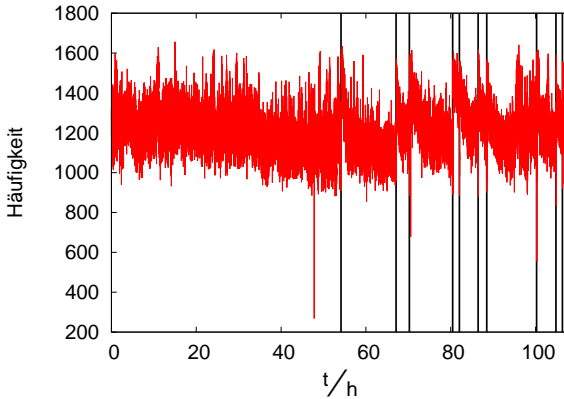


(a)

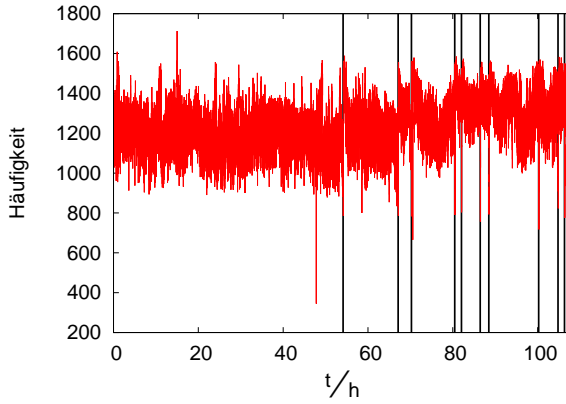
**Bild A.29:** Ergebnisse aus Simulation mit voller Gleitkommengenauigkeit mit Simulated Annealing für Kanal TL06. Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—).

### A.3 Epilepsievorhersage mit Musterdetektion

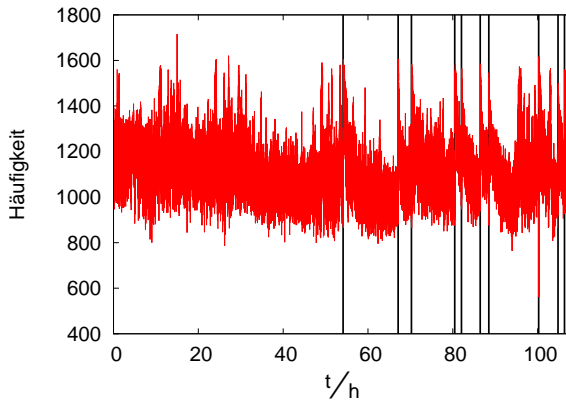
Es folgen weitere Ergebnisse zur Epilepsievorhersage mittels Musterdetektion.



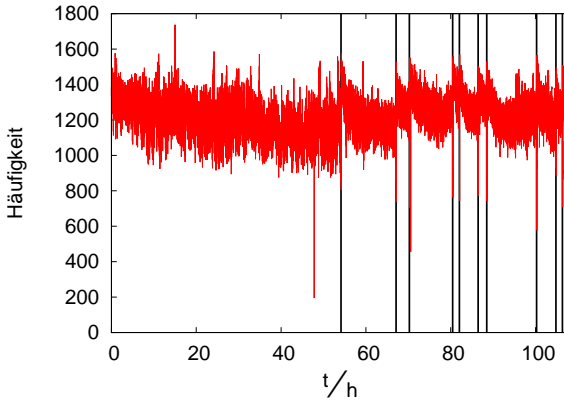
**Bild A.30:** Zeitliche Entwicklung der Musterhäufigkeiten in der Musterkategorie 0 für den Kanal TBAR4. Musterkategoriebildung über die Zeilen der Muster.



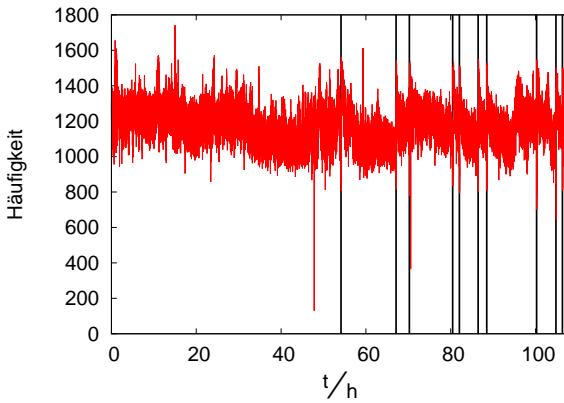
**Bild A.31:** Zeitliche Entwicklung der Musterhäufigkeiten in der Musterkategorie 0 für den Kanal TBPL04. Musterkategoriebildung über die Zeilen der Muster.



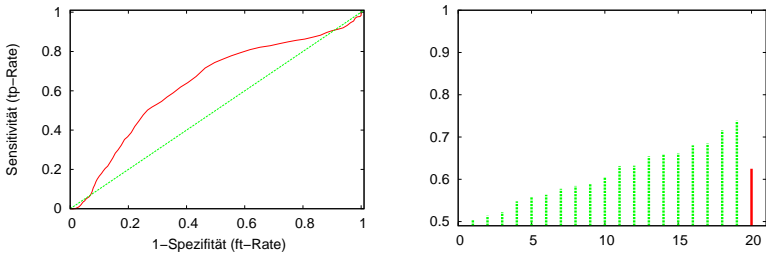
**Bild A.32:** Zeitliche Entwicklung der Musterhäufigkeiten in der Musterkategorie 0 für den Kanal TBPR4. Musterkategoriebildung über die Zeilen der Muster.



**Bild A.33:** Zeitliche Entwicklung der Musterhäufigkeiten in der Musterkategorie 0 für den Kanal TLL06. Musterkategoriebildung über die Zeilen der Muster.

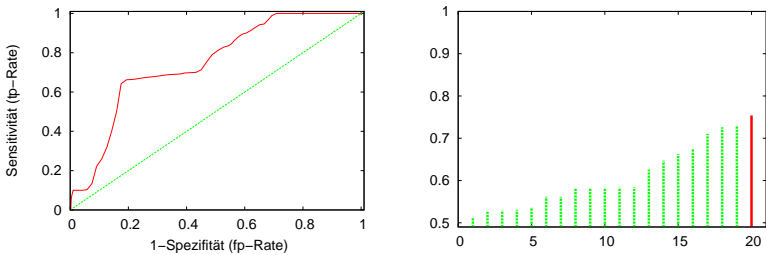


**Bild A.34:** Zeitliche Entwicklung der Musterhäufigkeiten in der Musterkategorie 0 für den Kanal TL06. Musterkategoriebildung über die Zeilen der Muster.



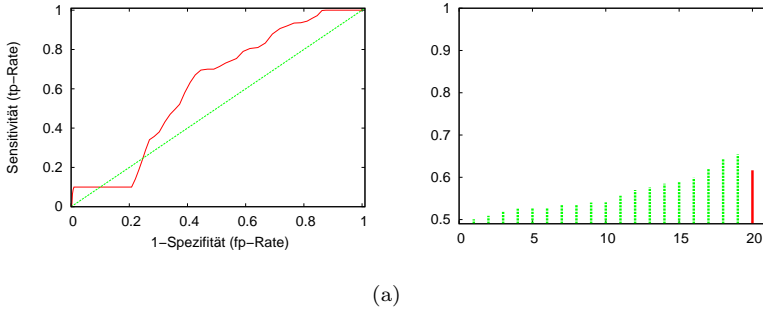
(a)

**Bild A.35:** Ergebnisse für die Sonde TBAR4. ROC Analyse für Musterkategorie 0 und Auszählrichtung vertikal (Korrelation der  $1 \times 3$  Muster). Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—) im EEG-Signal im Kanal.

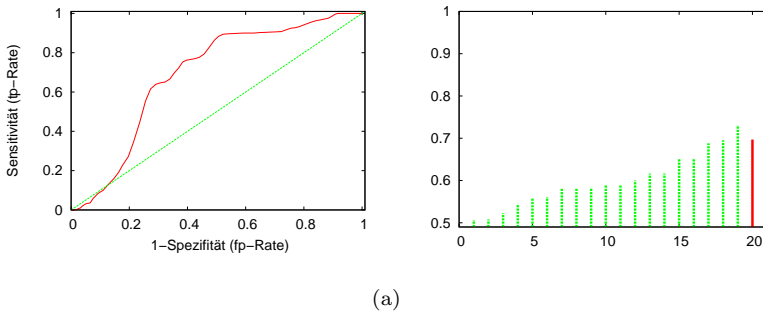


(a)

**Bild A.36:** Ergebnisse für die Sonde TBPL04. ROC Analyse für Musterkategorie 0 und Auszählrichtung vertikal (Korrelation der  $1 \times 3$  Muster). Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—) im EEG-Signal im Kanal.

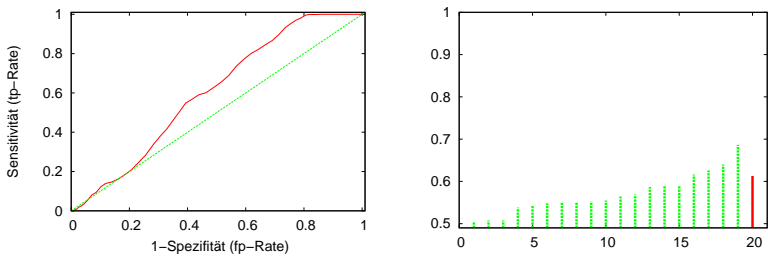


**Bild A.37:** Ergebnisse für die Sonde TBPR4. ROC Analyse für Musterkategorie 0 und Auszählrichtung vertikal (Korrelation der  $1 \times 3$  Muster). Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—) im EEG-Signal im Kanal.



**Bild A.38:** Ergebnisse für die Sonde TTL06. ROC Analyse für Musterkategorie 0 und Auszählrichtung vertikal (Korrelation der  $1 \times 3$  Muster). Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—) im EEG-Signal im Kanal.





(a)

**Bild A.39:** Ergebnisse für die Sonde TL06. ROC Analyse für Musterkategorie 0 und Auszählrichtung vertikal (Korrelation der  $1 \times 3$  Muster). Links: ROC-Kurven; rechts: dazugehöriger AUC-Wert, Linie bei 20 (—) und Ergebnisse der Surrogate, Linien bei 1 bis 19 (—) im EEG-Signal im Kanal.

## A.4 Key-Hole-Detektion mit digitalen Rechnern und Grafikprozessor

In diesem Teil des Anhangs wird eine weitere Untersuchung von Key-Hole Detektionsalgorithmen vorgestellt, die auf einer rein digitalen Implementierung des vorgestellten Algorithmus beruhen. Heutige Grafikkarten sind zumeist mit sehr leistungsfähigen Prozessoren, den Graphic Processing Units (GPU) ausgestattet. Diese sind für die parallele Verarbeitung relativ simpler Instruktionen ausgelegt. Es bietet sich an, auf solchen GPU's Bildverarbeitung durchzuführen, weshalb in den letzten Jahren immer mehr Hersteller dazu übergegangen sind, eine Application Programm Interface (API) für ihre Grafikkarten, z.B. CUDA von NVIDIA, anzubieten. Eine herstellerunabhängige API, OpenCL, wurde durch die CHRONOS Gruppe, ein Konsortium von Hardwareproduzenten, ins Leben gerufen. Diese ermöglicht es, alle in einem Rechnersystem verfügbaren „Compute-Einheiten“ zu nutzen, also auch die GPU auf der Grafikkarte. Eine weitere Möglichkeit der Parallelisierung bietet eine API, die unter dem Namen OpenMP (OMP) bekannt ist. Diese ermöglicht es, Programme in parallel ablaufende Prozesse aufzuteilen und auf Prozessoren zu verteilen.

Die digitale Implementierung des Key-Hole Detektionsalgorithmus wurde mit beiden Verfahren auf sein Laufzeitverhalten untersucht. Hierbei wurde eine auf die Grafikkarte angepasste Implementierung in zwei Versionen untersucht. Diese unterscheiden sich durch die Datenpartitionierung und Zuteilung auf die Prozessorkerne auf der GPU. Je nach Aufteilung hat dies merkliche Unterschiede im Laufzeitverhalten, wie in Tabelle A.6 angegeben, zur Folge. Die Implementierung mit OpenMP nutzt die oft mehrfach vorhandenen Prozessorkerne in den heute üblichen Prozessoren. Da die Daten nach der Berechnung wieder zusammengeführt werden müssen, kann hier eine

Verzögerung entstehen, da auf die Beendigung des langsamsten Berechnungsprozesses gewartet werden muß. Dies bedeutet für die Anwendung, Daten möglichst gleichmäßig auf die Kerne des Prozessors aufzuteilen. In den heutigen Rechnern sind zumeist Prozessoren mit zwei bis vier Kernen verbaut. Auf mindestens einem werden Prozesse des Betriebssystems ausgeführt, was zu einem Ungleichgewicht in der Auslastung der Kerne führen kann und damit zu unerwünschten Wartezeiten auf den langsamsten Prozess bis die Daten bearbeitet sind.

Verfahren	Laufzeit (ms)	Bilder/s
GPU mit openCL normal	1.61	6.22e+2
GPU mit openCL optimiert	0.562	1.78e+3
CPU mit openCL	0.852	1.17e+3
CPU mit OMP	0.159	6.29e+3
CPU mit Lookup Tabelle	0.248	4.03e+3
CPU optimiert	0.0810	1.23e+4
CPU Referenz	0.379	2.64e+003

**Tabelle A.6:** Laufzeiten für digitale Verfahren zur Key-Hole Detektion. Der auf CPU's optimierte Algorithmus ist schneller als der für openCL und GPU's optimierte Algorithmus

Der ohne Parallelisierung auf CPU's optimierte Algorithmus ist, wie in A.6 aufgeführt, schneller als die anderen hier aufgeführten. Das liegt vor allem daran, daß das Aufteilen der Daten im Falle von OpenMP als auch, im Falle von OpenCL (GPU's), sowie die Übertragung von Daten auf die Grafikkarte und zurück, bei Verwendung von GPU's, beträchtlich zur Gesamtlaufzeit beiträgt. Somit kann man für Bilder der Größe  $128 \times 128$  noch keinen Vorteil aus der parallelen Verarbeitung ziehen. Erst bei größeren Bildern wird sich eine Laufzeitverbesserung zugunsten von Multikernprozessoren bzw. GPU's ergeben, da dann die Zeit für die Datenaufteilung bzw. den Datentransfer in einem besseren Verhältnis zur Zeit, die die Berechnungen benötigen, stehen.



# Danksagung

Diese Dissertation entstand am Institut für Angewandte Physik der Johann Wolfgang Goethe-Universität Frankfurt am Main in der Arbeitsgruppe Nichtlineare Systeme und Signalverarbeitung von Herrn Prof. Dr. phil. nat. habil. Ronald Tetzlaff, bei dem ich mich für die Möglichkeit zur Anfertigung dieser Arbeit und seinen Ratschlägen, die zum Gelingen dieser Arbeit geführt haben, bedanken möchte. Meinen Dank möchte ich an dieser Stelle auch meinem Zweitgutachter Herrn Professor Dr. Arild Lacroix vor allem auch für die Betreuung in der letzten Phase der Promotion und Herrn Professor Dr. Maruhn für das Drittgutachten zollen. Außerdem gilt mein Dank Herrn Dr. Frank Gollas für die gute Zusammenarbeit in den vergangenen Jahren, in denen wir gemeinsam viele fachliche Diskussionen führten und Neues umgesetzt haben. Herrn Dipl.-Phys. Michael Reinisch danke ich für die Zeit, in der wir uns gemeinsam das tiefere Verständnis der Simulation von Zellularen Neuronalen Systemen und der Anwendung dieser auf Solitonen Lösungen für die Korteweg-DeVries- sowie der *Phi-4*-Gleichung erarbeitet haben. Auch möchte ich Frau Dr. Hermine Reichau, Herrn Dr. Müller, Herrn Dr. Martin Eichler und Familie Schiebe-Kötter für die Durchsicht und Anregungen danken, die zum Gelingen und Abrunden des Manuskripts beitragen. Mein Dank gilt auch Dr. Suat Suna, Dr. Roland Kunz, Dr. Frank Puffer, Dr. Christian Niederhöfer, Dipl.-Phys. Vanessa Senger, Dipl.-Phys. Lenica Reggie, Dipl.-Ing. Leonardo Nicolosie, Dipl.-Phys. Dennis Weiß, Dipl.-Phys. Andrej Loncar, Dipl.-Phys. Marc Geese und anderen Kollegen, sowie unseren Gästen Dr. Gergely Soos und Dr. Bela Weiß. Weiterhin möchte ich mich für die freundliche Arbeitsatmosphäre am Institut bei sämtlichen hier nicht nament-

lich genannten Mitarbeitern, Studenten und Gästen bedanken. Mein Dank gilt weiterhin Herrn Dipl.-Ing. Felix Abt und der Forschungsgesellschaft für Strahlwerkzeuge FGSW-Stuttgart für die Bereitstellung von Bildmaterial von Laserschweißvorgängen sowie Herrn Prof. Dr. Klaus Lehnertz vom Klinikum für Epileptologie, Bonn, und seiner Arbeitsgruppe für die zur Verfügungstellung von Patientendaten. Auch möchte ich mich für die vielfältige Unterstützung meiner Eltern, Geschwister, Freunde und Bekannten bedanken, die mir immer wieder Mut zugesprochen haben, der zur Fertigstellung der Arbeit geführt hat.

# Literaturverzeichnis

- [1] CHUA, L.O. und L. YANG: *Cellular Neural Networks: Theory and Applications*. IEEE Transactions on Circuits and Systems, 35:1257–1272, 1988.
- [2] DORMANN, S.: *Pattern Formation in Cellular Automation Models*. Dissertation, Universität Osnabrück, August 2000.
- [3] GRAY, L.: *A Mathematician Looks at Wolfram's New Kind of Science*. Notices of the Amer. Math. Soc., 50(2):200–211, 2003.
- [4] LONCAR, A.: *Kennlinienverfahren zur Modellierung nichtlinearer Systeme mit Zellularen Neuronalen Netzwerken*. Diplomarbeit, Institut für Angewandte Physik, Johann Wolfgang Goethe Universität Frankfurt am Main, Deutschland, 1999.
- [5] ANALOGIC AND NEURAL COMPUTING LABORATORY, COMPUTER AND AUTOMATION INSTITUTE, HUNGARIAN ACADEMY OF SCIENCESE, Budapest: *CNN Software Library (Templates and Algorithms)*, 1999. Version 7.3.
- [6] BRÄUNL, T., S.FEYER, W.PAPF und M.REINHARDT: *Parallele Bildverarbeitung*. Addison-Wesley, 1995.
- [7] CHUA, L. O. und T. ROSKA: *Cellular Neural Networks and visual computing, Foundation and applications*. Cambridge University Press, 2002.
- [8] CORINTO, F., M. GILLI und P.P. CIVALLERI: *On Stability of Full Range and Polynomial Type CNN*. In: *Proceedings of the 7<sup>th</sup> IEEE International Workshop of CNNs and their Applications (CNNA)*, pp. 33– 40, Frankfurt/Main, Germany, 2002.

- [9] CARMONA, R., F. JIMÉNEZ-GARRIDO, R. DOMÍNGUEZ-CASTRO, S. ESPEJO und A. RODRÍGUEZ-VÁSQUEZ: *CMOS Realisation of a 2-Layer CNN Universal Machine Chip*. In: TETZLAFF, PROF. DR. RONALD (Herausgeber): *Proceedings of the 7<sup>th</sup> IEEE International Workshop of CNNs and their Applications*, pp. 444–451. world Scientific, 2002.
- [10] LINÁN, G., A. RODRÍGUEZ-VÁSQUEZ, S. ESPEJO und R. DOMÍNGUEZ-CASTRO: *ACE16K: A 128 x 128 Focal Plain Analog Processor with Digital I/O*. In: TETZLAFF, RONALD (Herausgeber): *Proceedings of the International Workshop on Cellular Nonlinear Networks and their Applications*, pp. 132–139, Frankfurt, Germany, 2002.
- [11] LINÁN, G., R. DOMÍNGUEZ-CASTRO, S. ESPEJO und A. RODRÍGUEZ-VÁSQUEZ: *Design of a Large-Complexity Analog I/O CNNUC*. In: *Proceedings of the European Conference on Circuit Theory and Design (ECCTD)*, Band 40, pp. 42–57, 1999.
- [12] ANAFOCUS, LTD.: *ACE16kv2UG01*. Anafocus, Ltd., Anafocus Ltd. Avd. Isaac Newton s/n Pabellón de Italia, 7<sup>a</sup> Planta, Ático Parque Tecnológico Isla de la Cartuja 41092 Sevilla, Spain.
- [13] ANAFOCUS, LTD.: *Eye-RIS Vision System Hardware Description*. Anafocus, Ltd., Anafocus Ltd. Avd. Isaac Newton s/n Pabellón de Italia, 7<sup>a</sup> Planta, Ático Parque Tecnológico Isla de la Cartuja 41092 Sevilla, Spain.
- [14] ANAFOCUS, LTD.: *VISION SYSTEMS ON-CHIP - technology*. Homepage, Anafocus Ltd. Avd. Isaac Newton s/n Pabellón de Italia, 7<sup>a</sup> Planta, Ático Parque Tecnológico Isla de la Cartuja 41092 Sevilla, Spain, [www.anafocus.com](http://www.anafocus.com), 2005.
- [15] LAIHO, M., A. PAASIO, A. KANANEN und K. A. I. HALONEN: *A mixed-mode polynomial cellular array processor hardware realization*. *Circuits and Systems I: Regular Papers*, IEEE Transactions on, 51(2):286–297, Feb 2004.
- [16] LAIHO, M., A. PAASIO, A. KANANEN und K. HALONEN: *Realization of couplings in a polynomial type mixed-mode CNN*. In:



- 
- Proc. 7th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA 2002)*, pp. 422–429, 22–24 July 2002.
- [17] LAIHO, M., A. PAASIO und K. HALONEN: *Structure of a CNN cell with linear and second order polynomial feedback terms*. In: *Proc. 6th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA 2000)*, pp. 401–405, 23–25 May 2000.
- [18] AMES, C., D. FEIDEN, G. GEIS, R. KUNZ, A. LONCAR, F. PUFFER, R. SCHOENMEYER, R. TETZLAFF und D. WEISS: *SCNN 2000 Dokumentation*. Institut für Angewandte Physik, Johann Wolfgang-Goethe Universität Frankfurt am Main, 2000.
- [19] KUNZ, R., R. TETZLAFF und D. WOLF: *SCNN: A Universal Simulator for Cellular Neural Networks*. In: *Proceedings of the 4<sup>th</sup> IEEE International Workshop of CNNs and their Applications*, pp. 255–259, 06. 1996. CNNA-96, Seville, Spain.
- [20] LONCAR, A., R. KUNZ und R. TETZLAFF: *SCNN 2000 - Part I: Basic Structure and Features of the Simulation System for Cellular Neural Networks*. In: *Proceedings of the International Workshop on Cellular Neural Networks and their Applications*, Catania, 2000.
- [21] GEIS, G.C.: *Untersuchung von Parameterabweichungen auf das Verhalten von Zellularen Neuronalen Netzen*. Diplomarbeit, Institut für Angewandte Physik, Johann Wolfgang Goethe University Frankfurt am Main, Germany, 1998.
- [22] REGGIE, L.: *Bildkompression mit komplexen Systemen*. Diplomarbeit, Institut für Angewandte Physik, Johann Wolfgang Goethe-Universität Frankfurt am Main, 2008.
- [23] SENGER, V.: *Ionenstrahldiagnose mit Zellularen Neuronalen Netzen*. Diplomarbeit, Johann Wolfgang Goethe University Frankfurt am Main, 2009.

- 
- [24] GOLLAS, F.: *Identifikationsverfahren mit Reaktions-Diffusions Netzwerken zur Analyse hiernelektrischer Aktivität*. Dissertation, Johann Wolfgang Goethe University Frankfurt am Main, 2010.
- [25] STROUSTRUP, B.: *The C++ Programming Language, 3rd. ed.* Addison-Wesley, 1997.
- [26] ABADI, M. und L. CARDELLI: *A Theory of Objects*. Monographs in Computer Science. Springer-Verlag New York Berlin Heidelberg, 1995. ISBN 0-387-94775-2 Editors: D.Gries, F. B. Schneider.
- [27] BECK, K.: *Test Driven Development*. Addison-Wesley Longman, Amsterdam, 2002.
- [28] GAMMA, ERICH, RICHARD HELM, RALPH JOHNSON und JOHN VLISSIDES: *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman, Amsterdam, 1995.
- [29] STORN, R. und K. PRICE: <http://www.icsi.berkeley.edu/~storn/code.html>. Technischer Bericht.
- [30] GEIS, G., F. GOLLAS und R. TETZLAFF: *On the Implementation of Cellular Wave Computing Methods by Hardware Learning*. In: *Proceedings of SPIE's Microtechnologies for a New Millennium*, New Orleans, USA, 2007.
- [31] PRESS, W. H., B. P. FLANNERY, S. A. TEUKOLSKY und W.T. VETTERLING: *Numerical recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 2nd Edition Auflage, 1992.
- [32] ROSKA, T.: *Computational and computer complexity of analogic cellular wave computers*. In: *Proceedings of the 6<sup>th</sup> IEEE International Workshop of CNNs and their Applications, CNNA 2000*, Band ISBN 981-238-121-X, pp. 323–335, 2000.

- 
- [33] RODRÍGUEZ-VÁZQUEZ, A., G. LI NAN CEMBRANO, L. CARRANZA, E. ROCA-MORENO, R. CARMONA-GALÁN, F. JIMÉNEZ-GARRIDO, R. DOMÍNGUEZ-CASTRO und S.N ESPEJO MEANA: *ACE16k: The Third Generation of Mixed-Signal SIMD-CNN ACE Chips Toward VSoCs*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART I: REGULAR PAPERS Anafocus Ltd. Avd. Isaac Newton s/n Pabellón de Italia, 7<sup>a</sup> Planta, Ático Parque Tecnológico Isla de la Cartuja 41092 Sevilla, Spain, Vol. 51(No.5):851–863, 05. 2004.
- [34] ANAFOCUS, LTD.: *Eye-RIS Vision System evaluation kit Programmer's Guide*. Anafocus, Ltd., Anafocus Ltd. Avd. Isaac Newton s/n Pabellón de Italia, 7<sup>a</sup> Planta, Ático Parque Tecnológico Isla de la Cartuja 41092 Sevilla, Spain.
- [35] ANAFOCUS, LTD.: *Low Level instructions and functions User Guide*. Anafocus, Ltd., Anafocus Ltd. Avd. Isaac Newton s/n Pabellón de Italia, 7<sup>a</sup> Planta, Ático Parque Tecnológico Isla de la Cartuja 41092 Sevilla, Spain.
- [36] KUNZ, R. und R. TETZLAFF: *Evolutionary learning strategies for cellular neural networks*. In: *Proceedings of the 2000 6th IEEE International Workshop on Cellular Neural Networks and Their Applications, (CNNA 2000)*, pp. 241–246 (293–260), Catania, 2000. Evolutionary learning.
- [37] SOUZA, S. XAVIER DE, M. E. YALCIN, J. A. K. SUYKENS und J. VANDEWALLE: *Automatic Chip-Specific CNN Template Optimization Using Adaptive Simulated Annealing*. 2003. Krakow.
- [38] SOUZA, S. XAVIER DE, J.A.K. SUYKENS und J. VANDEWALLE: *Learning of Spatiotemporal Behavior in Cellular Neural Networks*. International Journal of Circuit Theory and Applications (CTA), 34(1):127–140, 2006.
- [39] DARWIN, C.: *On the Origins of Species by Means of Natural Selection*. John Murray, 1859.

- 
- [40] SCHÖNEBURG, E., F. HEINZMANN und F. FEDDERSEN: *Genetische Algorithmen und Evolutionsstrategien*. Addison-Wesley Verlag (Deutschland) GmbH, 1994.
- [41] STORN, R. und K. PRICE: *Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*. Technical Report TR-95-012, ICSI, March 1995 1995.
- [42] STORN, R. und K. PRICE: *Minimizing the real functions of the ICEC'96 contest by Differential Evolution*. In: *IEEE Conference on Evolutionary Computation, Nagoya*, pp. 842–844, 1996.
- [43] FAWCETT, T.: *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*. Intelligent Enterprise Technologies Laboratory HP Laboratories Palo Alto, HPL-2003-4:1–27, January 2003.
- [44] FAWCETT, T.: *An introduction to ROC analysis*. Pattern Recognition Letters, 27 (2006):861–874, 2005.
- [45] HANLEY, J.A. und B.J. MCNEIL: *The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve*. Radiology, 143(1):29–36, April 1982.
- [46] BRADLEY, A. P.: *The use of the area under the roc curve in the evaluation of mashine learning algorithms*. Pattern Recognition, 30:1145–1159, 1997.
- [47] KREUZ, T., R. G. ANDRZEJAK, F. MORMAN, A. KRASKOV, H. STÖGBAUER, C.E. ELGER, K. LEHNERTZ und P. GRASSBERGER: *Measure profile surrogates: A method to validate the performance of epileptic seizure prediction algorithms*. Phys Rev E, 69, 2004.
- [48] ANDRZEJAK, R. G., F. MORMANN, T. KREUZ, C. RIEKE, A. KRASKOV, C. E. ELGER und K. LEHNERTZ: *Testing the null hypothesis of the nonexistence of a preseizure state*. Physikal Review E, 67(1):010901, Jan 2003.

- 
- [49] SCHREIBER, T. und A.. SCHMITZ: *Surrogate time series*. Physica D: Nonlinear Phenomena, 142(3-4):346–382, 2000.
- [50] ENGBERT, R.: *Testing for nonlinearity: the role of surrogate data*. Chaos, Solitons & Fractals, 13(1):79–84, 2002.
- [51] TAYLOR, J. R.: *Fehleranalyse*. VCH Verlagsgesellschaft mbH, Weinheim, 1988.
- [52] WIKIPEDIA: *Binomialverteilung*. Technischer Bericht, 2012.
- [53] BARTLING, I.: <http://www.ingo-bartling.de/mathe/klasse12/html/stochastik/binomial/binomialvert.html>. Technischer Bericht.
- [54] GEESE, M.: *Kamerabasierte visuelle Regelung von Laserschweißprozessen mit Zellularen Neuronalen Netzwerken*. Diplomarbeit, J.W. Goethe Universität, 2008.
- [55] OPENOFFICE: <http://www.openoffice.org/de/>. Technischer Bericht, 2012.
- [56] OPENOFFICE: *CRITBINOM Funktion*. Technischer Bericht, 2012.
- [57] OPENOFFICE: *BINOMDIST Funktion*. Technischer Bericht, 2012.
- [58] REYNOLDS, EDWARD H und JAMES V KINNIER WILSON: *Psychoses of epilepsy in Babylon: the oldest account of the disorder*. Epilepsia, 49(9):1488–1490, 2008.
- [59] QUYEN, MICHEL LE VAN, FRÉDÉRIQUE AMOR und DAVID RUDRAUF: *Exploring the dynamics of collective synchronizations in large ensembles of brain signals*. J Physiol Paris, 100(4):194–200, Oct 2006.
- [60] BITTIGAU, P., M. SIFRINGER und C. IKONOMIDOU: *Antiepileptic Drugs and Apoptosis in the Developing Brain*. Annals of the New York Academy of Sciences, 993:103–114, 2003.
- [61] FLORIN, S.A.: *Zustandsklassifikation von nichtlinearen dynamischen Systemen mit Zellularen Neuronalen Netzen und mit*

- Untersuchung des Phasenskalierungsverhaltens.* Diplomarbeit, Rheinische Friedrich-Wilhelms-Universität Bonn, 2004.
- [62] LITT, B. und K. LEHNERTZ: *Seizure prediction and the preseizure period.* Current Opinion in Neurology, 15:173–177, 2002.
- [63] LOPES DA SILVA, F., B. WOUTER, N. KALITZIN STILIYAN, J. PARRA, P. SUFFCZYNSKI und D.N. VELIS: *Epilepsies as Dynamical Diseases of Brain Systems: Basic Models of the Transition Between Normal and Epileptic Activity.* Epilepsia, 44:72–83, 2003.
- [64] SCHELTER, B., J.TIMMER und A. SCHULZE-BONHAGE (Herausgeber): *Seizure Prediction in Epilepsy: From Basic Mechanisms to Clinical Applications.* Wiley-VCH, Berlin, 2008. ISBN 978-3-527-40756-9.
- [65] MORMANN, F., T. KREUZ., C. RIEKE, R. ANDRZEJAK, A. KRASKOV, P. DAVID, C. ELGER und K. LEHNERTZ: *On the predictability of epileptic seizures.* Clinical Neurophysiology, 116:569–587, 2005.
- [66] QUYEN, M. LE VAN, J. SOSS, V. NAVARRO, R. ROBERTSON, M. CHAVEZ, M. BAULAC und J. MARTINERIE: *Preictal state identification by synchronization changes in long-term intracranial EEG recordings.* Clinical Neurophysiology, 116(3):559–568, 2005 (March).
- [67] SCHINDLER, K., H. LEUNG, C.E. ELGER und K. LEHNERTZ: *Assessing seizure dynamics by analysing the correlation structure of multi-channel intracranial EEG.* Brain, 130:65–77, 2007.
- [68] KRUG, D., C.E. ELGER und K. LEHNERTZ: *A CNN-based synchronization analysis for epileptic seizure prediction: Inter- and intraindividual generalization properties.* In: *Proc. 11th International Workshop on Cellular Neural Networks and Their Applications CNNA 2008*, pp. 92–95, 14–16 July 2008.
- [69] NIEDERHÖFER, C., F. GOLLAS und R. TETZLAFF: *Prediction Error Profiles allowing a seizure forecasting in epilepsy?* In:

---

*CNNA 2006 10th International Workshop on Cellular Neural Networks and Their Applications*, Istanbul, Turkey, 2006.

- [70] GOLLAS, F., C. NIEDERHÖFER und R. TETZLAFF: *Prediction of Brain Electrical Activity in Epilepsy using a Higher Dimensional Prediction Algorithm for Discrete Time Cellular Neural Networks (DTCNN)*. In: *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04*, pp. V-720-V-723 Vol.5, Vancouver, Canada, 2004.
- [71] BADAWY, R., R. MACDONELL, G. JACKSON und S. BERKOVIC: *The peri-ictal state: cortical excitability changes within 24 h of a seizure*. *BRAIN a Journal of Neurology*, 132:1013-1021, 2009.
- [72] NIEDERHÖFER, C., F. GOLLAS und R. TETZLAFF: *Prediction of Brain Electrical Activity in Epilepsy by Multilayer Discrete Time Cellular Neural Networks (DTCNN)*. In: *Proceedings of the International Workshop on Cellular Nonlinear Networks and their Applications*, Budapest, Hungary, 2004.
- [73] NIEDERHÖFER, C. und R. TETZLAFF: *Recent Results on the Prediction of EEG Signals in Epilepsy by Discrete-Time Cellular Neural Networks (DTCNN)*. In: *Proceedings of the International Symposium on Circuits and Systems*, Kobe, Japan, 2005.
- [74] NIEDERHÖFER, C. und R. TETZLAFF: *Prediction of Epileptic Seizures using Multi-Layer Delay-Type Discrete Time Cellular Nonlinear Networks (DTCNN) - Long-Term Studies*. In: *Proceedings of SPIE's Microtechnologies for a New Millennium*, Sevilla, Spain, 2005.
- [75] TETZLAFF, R. und D. WEISS: *Cellular Neural Networks for the Anticipation of Epileptic Seizures*. In: *Proceedings of the International Symposium on Circuits and Systems*, Band 4 der Reihe *ISCAS 2002. IEEE International Symposium on Circuits and Systems, 2002*, pp. IV-177-IV-180, 2002-08-07 2002. SBN: 0-7803-7448-7.

- [76] BUSH, K. A.: *An Introduction to Adaptive Neurostimulation for the Treatment of Epilepsy*. Technischer Bericht, McGill University, Reasoning and Learning Lab, 2009.
- [77] XIZHENG, ZHANG, YIN LING und WANG WEIXIONG: *Wavelet Time-frequency Analysis of Electro-encephalogram (EEG) Processing*. International Journal of Advanced Computer Science and Applications, Vol.1(No. 5), 2010.
- [78] ENGEL, J.: *Seizures and Epilepsy*. F. A. Davis Company, Philadelphia, 1989.
- [79] FÜRNKRANZ, J. und P.A. FLACH: *An Analysis of Rule Evaluation Metrics*. Proceedings of the 20th International Conference on Machine Learning, 0:202–209, January 2003.
- [80] HANLEY, J.A. und B.J. MCNEIL: *A method of comparing the areas under receiver operating characteristic curves derived from the same cases*. Radiology, 148(3):839–843, September 1983.
- [81] KUNZ, R.: *Analyse hirnelektrischer Aktivität bei Epilepsie mit Zellularen Nichtlinearen Netzwerken*. Dissertation, Institut für Angewandte Physik, Johann Wolfgang Goethe University Frankfurt am Main, Germany, 2002.
- [82] DZAFIC, D.: *Analyse von EEG-Signalen mit Mustererkennungsverfahren bei Epilepsie*. Diplomarbeit, Institut für Angewandte Physik, Johann Wolfgang Goethe University Frankfurt am Main, Germany, 2010.
- [83] KUNZ, R., C. NIEDERHÖFER und R. TETZLAFF: *Prediction of epileptic seizures by CNN with linear weight functions*. In: *Proceedings of the International Workshop on Cellular Neural Networks and their Applications*, Frankfurt, Germany, 2002.
- [84] FISCHER, P. und R. TETZLAFF: *Pattern detection by cellular neuronal networks (CNN) in long-term recordings of a brain electrical activity in epilepsy*. Proceedings. 2004 IEEE International Joint Conference on Neural Networks (IJCNN), 1:169, July 25-29 2004.



- [85] MARTINI, H.: *Methoden der Signalverarbeitung*. Franzis Verlag GmbH, München, 1987.
- [86] MÜLLER-BORHANIAN, J.: *Integration optischer Messmethoden zur Prozess- kontrolle beim Laserschweißen*. Herbert Utz Verlag München, 2005.
- [87] ABT, DIPL.-ING. F.: *Dipl.-Ing. Felix Abt. Fachgespräch zum Kickoff-Meeting. 2007*.
- [88] MÜLLER-BORHANIAN, J., C. DEININGER, F. H. DAUSINGER und H. HÜGEL: *Spatially resolved on-line monitoring during LASER beam welding of steel and aluminum*. In: *Proceedings of the 23rd International Congress on Applications of Lasers & Electro-Optics*, 2004.
- [89] HÜGEL, H. und GRAF T.: *Laser in der Fertigung, Strahlquellen, Systeme, Fertigungsverfahren*. Vieweg + Teubner Verlag, 2009.



# Verzeichnis der Tabellen

2.1	Zellulare Systeme im Vergleich. . . . .	6
2.2	Ausgangsfunktionen von CNN. . . . .	17
3.1	Zeitmessung für C-Routinen. . . . .	34
3.2	Zeitmessung für FPP-Routinen. . . . .	35
5.1	Mittlere Anzahl von Netzwerkberechnungen pro Minute	60
6.1	Tabelle der vier Kombinationsmöglichkeiten für die ROC. . . . .	72
7.1	AUC-Werte mit konstantem Prädiktionshorizont. . . .	98
7.2	AUC-Werte mit variablem Prädiktionshorizont. . . .	98
7.3	AUC-Werte für Musterkategorie 0 . . . . .	111
7.4	Zeitmessung für einzelne Teilalgorithmen. . . . .	133
7.5	Zeitmessung für die digitalen Zählalgorithmen auf dem NIOS II Soft-Core-Prozessor. . . . .	134

- 7.6 Zeitmessung für die digitalen Zählalgorithmen auf einem PC (Pentium IV, 1.8 GHz). Im Gegensatz zum Programmablauf auf dem NIOS Prozessor des EyeRIS Systems laufen auf einem PC auch andere Programme, weshalb hier Zeitspannen der Laufzeiten angegeben werden. . . . . 134
  
- A.1 AUC Werte der verschiedenen Optimierungsverfahren für Kanal TBAR4. Aufgeführt sind die Ergebnisse für die Berechnungen auf dem EyeRIS-System und einer Simulation mit 8 bit Auflösung . . . . . 149
  
- A.2 AUC Werte der verschiedenen Optimierungsverfahren für Kanal TBPL4. Aufgeführt sind die Ergebnisse für die Berechnungen auf dem EyeRIS-System und einer Simulation mit 8 bit Auflösung . . . . . 152
  
- A.3 AUC Werte der verschiedenen Optimierungsverfahren für Kanal TBPR4 . . . . . 155
  
- A.4 AUC Werte der verschiedenen Optimierungsverfahren für Kanal TLL06 . . . . . 158
  
- A.5 AUC Werte der verschiedenen Optimierungsverfahren für Kanal TL06 . . . . . 161
  
- A.6 Laufzeiten für digitale Verfahren zur Key-Hole Detektion . . . . . 173

# Verzeichnis der Bildunterschriften

2.1	Die zwei am häufigsten angewendeten Nachbarschaften (graue Zellen zählen zur Nachbarschaft). . . . .	10
2.2	Einfluß einer Kopplungs-Template auf die Zelle in einem zweidimensionalen CNN (zweielementiger Indexvektor $[i, j]$ ) mit direkter Moore-Nachbarschaft ( $r = 1$ ) vgl. Bild 2.1(b). . . . .	10
2.3	Virtuelle Randzellen bei Moore-Nachbarschaft in einem zweidimensionalen CNN (zweielementiger Indexvektor $[i, j]$ ). Der Rand ist rot gekennzeichnet, die virtuellen Randzellen sind dunkelgrau unterlegt. . . . .	12
2.4	Mehrlagiges CNN; der Einflußbereich z.B. von Zelle 5 aus Lage 1 zu den Zellen der 2. Lage bei einer 3x3-Nachbarschaft ist kenntlich gemacht. . . . .	19
2.5	Verlustfunktion $g(x_i)$ des Full-Range Modells. Rot gestrichelt ist hier eine für Schaltungen realistische Kurvenform an den Umsprungspunkten der Funktion gezeigt. Die Bedeutung von $m$ in Formel 2.21 ist als Steigung der schwarz gestrichelten Linie dargestellt. . . . .	21
3.1	(a) EyeRIS-System ohne Optik (b) EyeRIS-System mit Optik und auf Halterung montiert. . . . .	26
3.2	Schematische Darstellung der Multiplizierereinheit des EyeRIS-Systems. . . . .	28

3.3	Zusammengefaßte und auf die Pixelanzahl normierte Statistik der Abweichungshäufigkeit . . . . .	37
4.1	Anordnung des symbolisierten Signals. . . . .	40
4.2	Shift- und UND-Operationen zur Erkennung eines im Bild enthaltenen Musters. Grau dargestellt ist die ursprüngliche Position der Pixel im Bild. Pfeile zeigen in Richtung der Verschiebung. . . . .	42
4.3	Shift- und UND-Operationen zur Erkennung eines nicht im Bild enthaltenen Musters. Grau dargestellt ist die ursprüngliche Position der Pixel im Bild. Pfeile zeigen in Richtung der Verschiebung. . . . .	43
4.4	Mögliche Detektion von Mustern die nicht im Bild enthalten sind. . . . .	44
4.5	Mögliche Detektionsfehler durch in das Netz Hineinverschieben von Werten aus den virtuellen Randzellen. . . . .	45
4.6	Musterbild 170. . . . .	46
4.7	Muster-zu-Musterbild-Fehler: erst Bild dann Muster. . . . .	48
4.8	Fehlerhafte Zellen im Chip. . . . .	49
4.9	Muster-zu-Musterbild-Fehler: erst Muster dann Bild. . . . .	50
5.1	SSAPE-Flußgraph. . . . .	56
5.2	Beispiel einer Symmetrieoperation: Spiegelung an der Mittelachse. . . . .	58
5.3	Zeitliche Entwicklung des MSE. . . . .	62
5.4	Beispiel für die Hole-Filler-Verarbeitung. . . . .	63
5.5	Beispiel für die Kantendetektion. . . . .	65

---

5.6	Limited Diffusion. . . . .	67
5.7	Fehlerentwicklung für begrenzte Diffusion. . . . .	68
6.1	Signal mit Schwellwert. . . . .	72
6.2	Verteilungen der $TP$ , $TN$ , und die Überschneidung aus der sich zusammen mit dem Schwellwert $FP$ und $FN$ ergeben. . . . .	73
6.3	Area Under the Curve (AUC). . . . .	74
7.1	Für eine Prädiktion eines epileptischen Anfalls wich- tige Zeitbereiche in einem EEG-Signal. . . . .	81
7.2	Erzeugung von Surrogaten durch Verschieben von be- kannten Anfallszeitpunkten. . . . .	83
7.3	Erzeugung von Surrogaten durch Austausch von Da- tensegmenten. . . . .	83
7.4	Implantationsschema. . . . .	86
7.5	Beispiel für ein EEG-Signal. . . . .	87
7.6	Modellierungsschema. . . . .	90
7.7	Umsetzung des Delays auf dem EyeRIS-System. . . . .	91
7.8	SSAPE nach 30 Iterationen. . . . .	93
7.9	Entwicklung des Prädiktionsfehlers mit SSAPE über 30 Iterationen. . . . .	94
7.10	SSAPE bei 5 facher Wiederholung und Mittelung. . . . .	95
7.11	Typische Ergebnisse für die Fehlerwerte als Funktion der Zeit für DT-CNN simuliert mit 8 bit Genauigkeit. . . . .	99
7.12	ROC-Kurven und dazugehörige AUC-Werte . . . . .	100

7.13	Symbolisierung. . . . .	101
7.14	Symbolisierung. . . . .	102
7.15	Anordnung des symbolisierten Signals. . . . .	103
7.16	Symbolisierung. . . . .	104
7.17	Anordnung für 2D Muster ohne Berücksichtigung der Aufspaltung. . . . .	105
7.18	Anordnung für 2D Muster mit Berücksichtigung der Aufspaltung. . . . .	106
7.19	Lineare Musterdetektion. . . . .	107
7.20	Bildung von Musterklassen. . . . .	109
7.21	Musterklassenüberblick. . . . .	110
7.22	Musterklassifikation. . . . .	111
7.23	Musterklassifikation. . . . .	112
7.24	Zwei typische Ergebnisse der ROC Analyse für Mu- sterkategorie 0 und vertikaler Auszählrichtung. . . . .	113
7.25	Ergebnisse der Musterdetektion für Kategorie 0. . . . .	114
7.26	Querschnittschema einer Durchschweißung. . . . .	115
7.27	Einkopplung der Laserleistung . . . . .	117
7.28	Koaxiale Aufnahme durch den Strahlengang des Lasers	119
7.29	Key-Hole Erkennung mit einem Schwellwert . . . . .	121
7.30	Key-Hole nicht detektierbar . . . . .	122
7.31	Key-Hole detektierbar . . . . .	123
7.32	Anwendung von XOR. . . . .	124
7.33	CNN Algorithmus zur Erkennung des Key-Holes. . . . .	125



---

7.34	Anwendung der Schwellwerte. . . . .	126
7.35	Auszählen der schwarz/weiß-Übergänge. . . . .	127
7.36	Anwendung einer Maske. . . . .	129
7.37	Lücken, die nach der Masken-Operation. . . . .	131
7.38	Schwellwertbilder. . . . .	136
7.39	Guter Schweißprozess. . . . .	137
7.40	Guter Schweißprozess. . . . .	138
7.41	Schweißprozess mit Fluktuation der Helligkeit . . . . .	139
7.42	Schlechter Schweißprozess . . . . .	140
A.1	Anwendung der Hole-Filler-Verarbeitung. . . . .	145
A.2	Anwendung der Hole-Filler-Verarbeitung bei mehreren Objekten. . . . .	146
A.3	Fehlerentwicklung während der Optimierung bei der Hole-Filler-Verarbeitung. . . . .	146
A.4	Anwendung der Edge-Detection-Verarbeitung bei mehreren Objekten. . . . .	147
A.5	Fehlerentwicklung während der Optimierung bei der Hole-Filler-Verarbeitung. . . . .	148
A.6	SSAPE auf Eye-RIS System für Kanal TBAR4 . . . . .	149
A.7	DE auf Eye-RIS System für Kanal TBAR4 . . . . .	150
A.8	SSAPE TBAR4 8 bit . . . . .	151
A.9	DE TBAR4 8 bit . . . . .	151
A.10	SSAPE auf Eye-RIS System für Kanal TBPL4 . . . . .	152
A.11	DE auf Eye-RIS System für Kanal TBPL4 . . . . .	153

---

A.12 SSAPE TBPL4 8 bit . . . . .	154
A.13 DE TBPL4 8 bit . . . . .	154
A.14 SSAPE auf Eye-RIS System für Kanal TBPR4 . . . . .	155
A.15 DE auf Eye-RIS System für Kanal TBPR4 . . . . .	156
A.16 SSAPE TBPR4 8 bit . . . . .	157
A.17 DE TBPR4 8 bit . . . . .	157
A.18 SSAPE auf Eye-RIS System für Kanal TLL06 . . . . .	158
A.19 DE auf Eye-RIS System für Kanal TLL06 . . . . .	159
A.20 SSAPE TLL06 8 bit . . . . .	160
A.21 SSAPE TLL06 8 bit . . . . .	160
A.22 SSAPE auf Eye-RIS System für Kanal TL06 . . . . .	161
A.23 DE auf Eye-RIS System für Kanal TL06 . . . . .	162
A.24 DE TL06 8 bit . . . . .	163
A.25 DE TL06 8 bit . . . . .	163
A.26 SSAPE TL06 . . . . .	164
A.27 DE TL06 . . . . .	164
A.28 BFGS TL06 . . . . .	165
A.29 Simulated Annealing TL06 . . . . .	165
A.30 Ergebnisse der Musterdetektion der Kategorie 0 für Kanal TBAR4. . . . .	166
A.31 Ergebnisse der Musterdetektion der Kategorie 0 für Kanal TBPL04. . . . .	167
A.32 Ergebnisse der Musterdetektion der Kategorie 0 für Kanal TBPR4. . . . .	167

---

A.33 Ergebnisse der Musterdetektion der Kategorie 0 für Kanal TTL06. . . . .	168
A.34 Ergebnisse der Musterdetektion der Kategorie 0 für Kanal TL06. . . . .	168
A.35 Ergebnisse für die Sonde TBAR4. . . . .	169
A.36 Ergebnisse für die Sonde TBPL04. . . . .	169
A.37 Ergebnisse für die Sonde TBPR4. . . . .	170
A.38 Ergebnisse für die Sonde TTL06. . . . .	170
A.39 Ergebnisse für die Sonde TL06. . . . .	171



# LEBENS LAUF

Gunter C. Geis

geboren am 5. Februar 1971

in Frankfurt am Main

1978–1991	Freie Waldorfschule Frankfurt am Main
1991	Erlangung der allgemeinen Hochschulreife
1991–1992	Zivildienst in einer Altenwohnanlage
1992–1998	Studium der Physik an der Johann Wolfgang Goethe-Universität Frankfurt am Main (Vordiplom 1994) Abschluß Diplom
1998	Systemadministration, NSC (Frankfurt am Main)
1998–1999	Softwareentwicklung, Test und Portierung, Cetec (Bad Homburg v.d.H.)
2000	Softwareentwickler für EUREX Handelsplattform, Deutsche Börse (Frankfurt am Main)
2001	Systemadministration für Firmen im Handelsgewerbe (Inneneinrichtungen und Internethandel)
2001	Annahme als Doktorand am Institut für Angewandte Physik der J. W. Goethe-Universität
2002–2007	Wissenschaftlicher Mitarbeiter an der J.W. Goethe-Universität Frankfurt am Main
2007	Softwareentwicklung für eine Börsendaten-Handelsplattform, Tenfore (Frankfurt am Main)
2007–2008	Softwareentwicklung in der Eisenbahnsicherheitstechnik, Firma Signal und System Technik (Siershan)
2009	Webentwickler, Adacor (Offenbach am Main)
2010	Selbständiger IT-Consultant
2011	Aufbau der Abteilung Softwareentwicklung mobile Geräte Firma CBC GmbH

Veröffentlichungen:

- R. Tetzlaff, R. Kunz, G. Geis; Analysis of Cellular Neural Networks with Parameter Deviations, Proceedings of IEEE ECCTD 97, Budapest, pp.650-654, 1997
- R. Tetzlaff, R. Kunz, G. Geis; Minimizing the Effects of Tolerance Faults on Hardware Realizations of Cellular Neural Networks, CNNA 98, London, pp 385-390, 1998
- G. Geis, M. Reinisch, R. Tetzlaff; Linear Interpolation of Nonlinearities in Cellular Neural Networks (CNN), Proc. CNNA2004 8th IEEE International Workshop on Cellular Neural Networks and their Applications, Budapest, ISBN 963 311 357 1, pp. 393-398
- G. Geis, D. Feiden, R. Tetzlaff; Recent results for obstacle detection by Cellular Neural Networks(CNN), SPIE05, Seville, Mai 2005
- M.Reinisch, G. Geis, R. Tetzlaff; System Identification by Cellular Neural Networks (CNN): Linear Interpolation of Nonlinear Weight Functions, SPIE05, Seville, Mai 2005
- G.Geis, V. Senger, R. Tetzlaff; Implementation of cellular wave computing methods by hardware learning: ion beam analysis, Cellular Neural Networks and Their Applications CNNA, 2006 10th International Workshop on Cellular Neural Networks and their Applications, 2006
- G. Geis, F. Gollas, R.Tetzlaff; Optimierungsverfahren für schaltungstechnische Realisierungen zellularer Rechner, Kleinheubacher Tagungen, Miltenberg, 2006
- G.Geis, F. Gollas, R. Tetzlaff; Niederhöfer, C., Gollas, F. and Tetzlaff, R., On the Implementation of Cellular Wave Computing Methods by Hardware Learning, (DTCNN), ISCAS 2007
- G.Geis, F.Gollas, and R.Tetzlaff, Towards an automated seizure anticipation device based on Cellular Neural Networks

(CNN), International Workshop on Cellular Neural Networks  
and their Applications 2010, CNNA 2010

Akademische Lehrer in alphabetischer Reihenfolge:

Prof. Dr. H.F. de Groote	Prof. Dr. Th. Elze
Prof. Dr. H. Klein	Prof. Dr. J. Kummer
Prof. Dr. A. Lacroix	Prof. Dr. W. Martienssen
Prof. Dr. E. Mohler	Prof. Dr. H. Reininger
Prof. Dr. A. Schaarschmidt	Prof. Dr. A. Schemp
Prof. Dr. H. Schmidt-Böcking	Prof. Dr. H. Stöcker
Prof. Dr. R. Tetzlaff	Prof. Dr. K. Waldschmidt
Prof. Dr. D. Wolf	