

Bildererkennung mit Hilfe des Neocognitrons

An der Fachhochschule Dortmund
im Fachbereich Informatik
erstellte Diplomarbeit
im Studiengang Technische Informatik
[Technische Informatik]

zur Erlangung des Grades Diplom-Informatiker (FH) von

Mirco Ehrich
Thorsten Schröder

Betreuer: Prof. Dr. Eberhard von Goldammer

Dortmund, 24. Februar 2004

Danksagung

Wir möchten und ganz herzlich bei Herrn Prof. Dr. von Goldammer für seine Geduld und Unterstützung bedanken. Ebenfalls möchten wir uns bei Herrn Hofmann bedanken. Er hat uns mit Rat und Tat zur Seite gestanden, wenn wir Probleme mit unserem Computer hatten.

Inhaltsverzeichnis

1 Einleitung.....	5
2 Fuzzy Logik	7
2.1 Einführung.....	7
2.2 Fuzzy-Mengen (Fuzzy-Set).....	9
2.3 Fuzzy in der Bildverarbeitung.....	10
2.4 Bilder als Menge	11
2.5 Kantendetektion mit lokaler Fuzzyfizierung	13
3 Das Visuale System.....	19
3.1 Einführung.....	19
3.2 Grundlagen.....	21
3.2.1 Die Retina	21
3.2.2 Das rezeptive Feld	23
3.2.3 Die Simple Cell.....	25
3.3 Die Complex Cell.....	27
3.3.1 Die Lower-order hypercomplex Cell.....	28
3.3.2 Die Higher-order hypercomplex Cell.....	29
3.4 Nachtrag.....	30
4 Aufbau und Arbeitsweise des Neocognitrons.....	31
4.1 Einführung.....	31
4.2 Das Cognitron.....	33
4.3 Das Neocognitron	34
4.4 Die Eingangsinformationen des Neocognitrons	35
4.5 Erster Layer simple cells.....	35
4.6 Erster Layer Complex Cell.....	39
4.7 Layer 2 und folgende.....	40
4.8 Training des Neocognitrons.....	46
4.8.1 Lernen ohne Lehrer.....	46
4.8.2 Lernen mit Lehrer.....	47
4.9 Die Morphologie des Neocognitrons.....	52
4.10 Die synaptische Organisation des Neocognitrons	53

4.11 Die mathematische Beschreibung des Neocognitrons.....	55
5 Anwendung des Neocognitron in dieser Arbeit.....	57
5.1 Der Aufbau.....	57
5.2 Die Trainingsmethoden.....	59
5.3 Die Parameter des Neocognitrons.....	59
5.4 Das Anlernen und Erkennen von dreidimensionalen Mustern	60
5.4.1 Aufbau der Ergebnistabellen	60
5.4.2 Ergebnisse: Toleranz.....	61
5.4.3 Ergebnisse: Lerndurchgänge und Lernfaktor.....	64
5.5 Das Anlernen und Erkennen von zweidimensionalen Mustern.....	66
5.6 Ergebnisse : Translation und Deformation	68
6 Ausblick	70
7 Zusammenfassung	70
8 Anhang A: Die Programmbedienung.....	72
8.1 Hardwareanforderungen.....	72
8.2 Programmübersicht.....	72
8.2.1 Kamera Initialisierung.....	73
8.2.2 Lernverfahren.....	74
8.2.3 Lernen ohne Lehrer.....	75
8.2.4 Lernen mit Lehrer.....	76
8.2.5 Lernen (Layer 1 mit Lehrer / Rest ohne Lehrer).....	80
8.2.6 Erkennen.....	81
8.2.7 Graustufen.....	82
9 Anhang B: Abbildungsverzeichnis.....	92
10 Anhang C: Tabellenverzeichnis.....	94
11 Anhang D: Literaturverzeichnis.....	95

1 Einleitung

Das Neocognitron gilt als eines der kompliziertesten Neuronalen Netze. Der Grund liegt in der starken Orientierung des Neocognitrons an dem „Visuellen System“ der Säugetiere. Wegen dieser Kompliziertheit wurde das Neocognitron lange vernachlässigt und es existieren hierüber nur wenige Arbeiten in der Literatur.

Ein zusätzlicher Grund ist möglicherweise eine von dem Entwickler des Neocognitrons, Fukushima, selbst publizierte Anwendung. Sein Neocognitron, Anfang der achtziger Jahre zur Handschriftenerkennung der Zahlen von 0 bis 9 entwickelt, war eines der ersten und wenigen Neuronalen Netze, die auf dem Neocognitron basieren. Durch die damalige noch sehr begrenzte Computerkapazität war die Darstellungsgröße der zu erkennenden und erlernenden Zahlen sehr beschränkt und auch das Anlernen der Zahlen dauerte sehr lang und war sehr aufwendig. Durch die Darstellung immer nur dieses einen Beispiels in der Literatur erhielt das Neocognitron seinen Ruf, unflexibel, langsam und sehr kompliziert zu sein.

Im Rahmen dieser Diplomarbeit wurde ein Neocognitron programmiert, das die Möglichkeit besitzt, allgemeinere und größere Muster als im Beispiel von Fukushima zu erlernen und zu erkennen. Es sollte damit untersucht werden, welches Potential und welche Flexibilität das Neocognitron besitzt.

Die Diplomarbeit hat folgenden Aufbau:

Im zweiten Kapitel erfolgt eine kurze Einführung in die Fuzzy Logik und eine Beschreibung der Funktionsweise einer Kantendetektion mit lokaler Fuzzyfizierung. Die Kantendetektion dient dabei zu Aufbereitung von Bildern, um sie später mit dem Neocognitron verarbeiten zu können.

Das dritte Kapitel gibt einen kurzen allgemeinen Überblick über die Neuronalen Netze und den Aufbau des „Visuellen Systems“ von Säugetieren. Um ein Verständnis für die Arbeitsweise des Neocognitrons aufzubauen wird für die Beschreibung des „Visuellen Systems“ ein ganzer Abschnitt eingeräumt.

Im vierten Kapitel erfolgt dann die Erläuterung über den Aufbau und die Arbeitsweise des Neocognitrons selbst.

Im fünften Kapitel werden die Ergebnisse dieser Diplomarbeit dargestellt und Diskutiert.

Abschließend erfolgt dann eine kurze Zusammenfassung und ein Ausblick.

Im Anhang findet sich eine Anleitung zur Bedienung des Neocognitrons welches im Rahmen dieser Diplomarbeit programmiert wurde.

2 Fuzzy Logik

2.1 Einführung

Was ist eine große Zahl und ab wann ist eine Zahl sehr groß?

Eigentlich eine einfache Frage. Wenn die Frage aber genauer betrachtet wird, ist sie nicht mehr ganz so einfach zu beantworten, denn es gibt keinen genau definierten Übergang, ab wann Zahlen groß sind und was der Unterschied zwischen großen und sehr großen Zahlen ist. Es gibt somit auch keine exakte Antwort.

Würde man eine Menge der großen Zahlen erstellen, so müßte als erstes eine exakte Grenze definiert werden, ab wann eine Zahl „groß“ ist und ab wann sie „sehr groß“ wird. Diese Definition ist dann allerdings rein willkürlich geschehen.

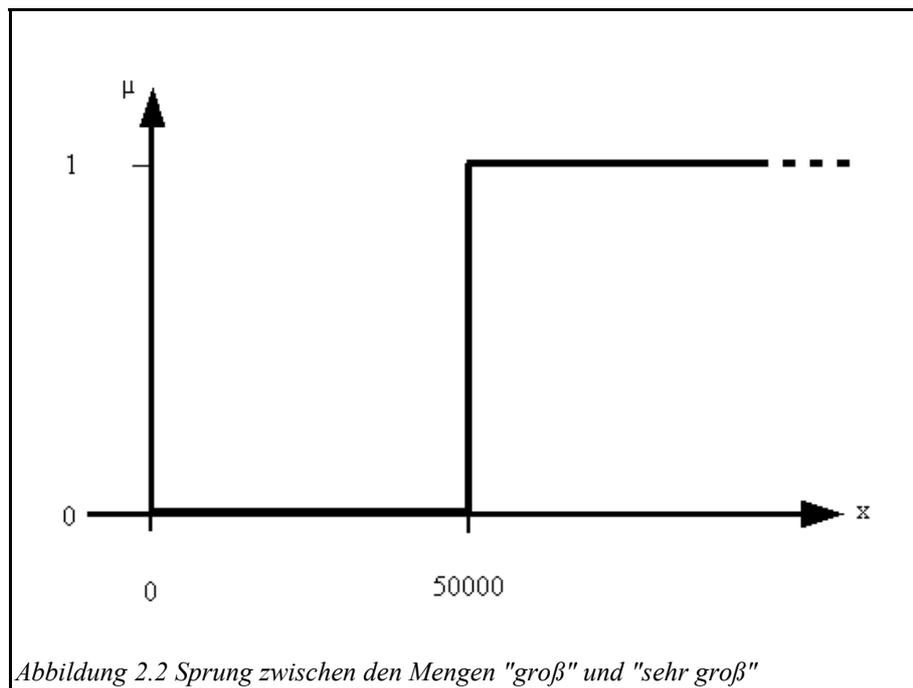
Betrachtet man eine willkürlich festgelegte Menge M zwischen „großen“ reellen Zahlen ab 500 bis zum Beginn von „sehr großen“ Zahlen bei 50000

$$M := \{x | x \in \mathbb{R}, 500 \leq x < 50000\} \quad (1)$$

dann fällt auf, daß $x = 499.999$ kein Element der Menge M der großen Zahlen ist, $x = 500$ aber zur Menge gehört und ebenso, daß $x = 49999$ eine große Zahl ist, $x = 50000$ aber schon zur Menge der „sehr großen“ Zahlen gehören würde. Dies wird in Abbildung 2.1 schematisch dargestellt.



Ein Element der Menge M kann entweder zur Menge gehören, also den Zugehörigkeitsgrad „wahr“ und damit die Zugehörigkeitsfunktion $\mu(x) = 1$ haben, oder nicht zur Menge M gehören und damit den Zugehörigkeitsgrad „falsch“, $\mu(x) = 0$ haben. Es gibt aber keine Zwischenzustände. Deswegen kann es zu dem seltsamen Effekt kommen, daß Unterschiede in umgangssprachlichen Begriffen wie „groß“ zu „sehr groß“ (Abbildung 2.2) in der klassischen Mengenlehre abrupt als Sprung dargestellt werden.



Besser wäre es, einen stetigen Übergang, einen Bereich, statt eines Sprungs zu benutzen, um dem fließenden Übergang zwischen „groß“ und „sehr groß“ besser gerecht zu werden. Allerdings ist dieses nicht mehr mit Hilfe der klassischen zweiwertigen Logik (wahr oder falsch) darstellbar.

2.2 Fuzzy-Mengen (Fuzzy-Set)

Bereits im Jahre 1913 erweiterte der polnische Logiker Jan Lukasiewicz [lu13] die zwei möglichen Zustände „wahr“ („0“) und „falsch“ („1“) um den weiteren Zustand „nicht entscheidbar“. Nach seiner Betrachtungsweise existiert dieser Zustand dann, wenn die Zustände „wahr“ und „falsch“ gleichzeitig vorliegen („01“ oder „10“).

Der eigentliche Durchbruch gelang Lotfi A. Zadeh [za65] mit der Entwicklung der „unscharfen Mengen“ (Fuzzy-Mengen). Er erweiterte den Ansatz von Lukasiewicz auf das gesamte reelle Intervall $[0,1]$ mit quasi unendlich vielen Zwischenzuständen, wobei M eine Menge ist und ihre Abbildung

$$\mu : M \rightarrow [0,1] \quad (2)$$

eine unscharfe Menge (Fuzzy-Menge) in M darstellt. Dabei ist μ die Zugehörigkeitsfunktion der Fuzzy-Menge, die jedes Element $x \in M$ einem Zugehörigkeitsgrad $\mu(x)$ aus dem Intervall $[0,1]$ zuordnet.

Auf die Menge der großen Zahlen aus dem Anfangsbeispiel bezogen heißt dies, daß die Grenzen einer klassischen Menge unscharf dargestellt werden müssen. Es wird also für den Grenzbereich ein Punkt benötigt, der minimal zur Menge gehört und ein Punkt, der komplett zur Menge der großen Zahlen gehört. Ein Beispiel ist :

Der Zugehörigkeitsgrad für $x < 100$ zur Menge ist 0.

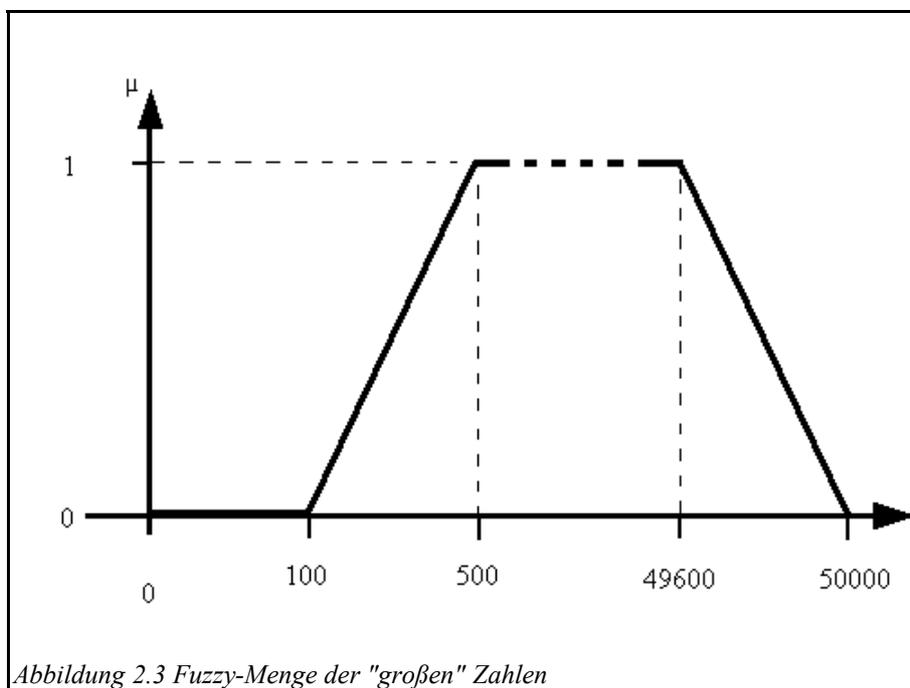
Der Zugehörigkeitsgrad für $x = 100$ zur Menge ist 0,1.

Der Zugehörigkeitsgrad für $500 < x < 49600$ zur Menge ist 1.

Der Zugehörigkeitsgrad für $x = 50000$ zur Menge ist 0,1

Der Zugehörigkeitsgrad für $x > 50000$ zur Menge ist 0.

Die daraus folgende Menge ist eine Fuzzy-Menge oder unscharfe Menge (Abbildung 2.3).



Im allgemeinen werden keine willkürlichen Grenzwerte für eine Fuzzy-Menge benutzt, vielmehr ergeben sich deren Werte z.B. aus statistischen Erhebungen, Messungen oder Berechnungen.

2.3 Fuzzy in der Bildverarbeitung

Wenn ein Mensch ein Bild betrachtet, z.B. ein Foto oder Gemälde, erkennt er automatisch was wichtige und was unwichtige Informationen auf diesem Bild sind. Er kann den Vordergrund vom Hintergrund separieren und es ist ihm möglich, Umrisse und die dazu gehörenden Objekte richtig zu erkennen, auch wenn sie halb verdeckt sind.

Das Erkennen von Objekten mit einem Computer ist nicht so einfach möglich. Der Computer, oder vielmehr das eingesetzte Programm, hat keine Vorstellung was ein Baum, ein Haus oder ein Auto ist. Was für den Menschen sehr einfach ist, stellt den Computer vor eine fast unüberwindliche Hürde. Dem Menschen ist es möglich, aus einer Ansammlungen von Farben und Helligkeitsverteilungen Gruppen (oder Mengen) heraus zu finden die zusammengehören und damit auch Objekte zu erkennen. Dieses beruht darauf, daß das Aussehen dieser Objekte erlernt wurde, also auf Erfahrungen zurückgegriffen werden konnte.

Einem Computer stehen auch diese Farb- oder Helligkeitsverteilungen eines Bildes zur Verfügung,

allerdings kann er nicht auf ein erlerntes Wissen zurückgreifen um eine Gruppierung von solchen Bildpunkten vorzunehmen. Ihm muß mit einem mathematischen Formalismus erklärt werden was zusammengehört.

2.4 Bilder als Menge

Betrachtet man ein Bild an einem Monitor so stellt man fest, daß es aus Bildpunkten (Pixel) zusammengesetzt ist. Mit Computer bearbeitete Bilder können z.B. eine Farbtiefe von 24 Bit haben, d.h. sie besitzen 16,7 Millionen Farbtönen (Abbildung 2.4).



Die Farbe eines Bildpunktes wird dabei aus den drei Farben rot, grün und blau (RGB) zusammengesetzt. Jede dieser Farben besitzt die Möglichkeit, 256 verschiedene Helligkeitsabstufungen (8 Bit) in ihrer Farbe anzunehmen und zwar von 0 (sehr dunkel) bis 255 (sehr hell). Damit ergibt sich dann ein Farbraum von 24 Bit.

Die Frage ist nun was man mit diesen Informationen anfangen kann. Um Objekte in einem Bild zu erkennen, reichen Farbe und Helligkeitsinformationen nicht aus. Objekte in einem Bild bestehen selten nur aus einer Farbe oder einer Helligkeit, es ist also nicht möglich, einfach die verschiedenen Farben als Mengen anzusehen oder die Intensitäten einer Farben als den Zugehörigkeitsgrad zu einer Menge zu interpretieren.

Eine bessere Möglichkeit ist es die Umrisse, also die Kanten von Objekten, als eine oder mehrere Mengen anzusehen und für jeden Pixel den Zugehörigkeitsgrad zu dieser Menge, also der Kante,

anzugeben. Dafür werden nur die Helligkeitsinformationen benutzt, da eine Kante sich dadurch auszeichnet, daß es einen mehr oder weniger plötzlichen Übergang zwischen hellen und dunklen Bildinformationen gibt. Für die Helligkeitsinformationen in einem Bild reicht ein Graustufenbild (Abbildung 2.5), da in diesem schon alle Farbwerte in Helligkeitswerte umgerechnet sind.



Abbildung 2.5 Graustufenbild 8-Bit

2.5 Kantendetektion mit lokaler Fuzzifizierung

Eine Kantendetektion ist die Erkennung von Objekt-Umrissen in einem Graustufenbild. Dabei ist die Detektion einer Kante in einem weiten Bereich möglich, von einer nur leichten Hervorhebung der Kanten bis zu deren vollständigen Separierung. Im letzteren Fall besteht das Bild nur noch aus den Kanten der Objekte, die im Bild dargestellt sind.

Bei der lokalen Fuzzifizierung wird das Bild in Ausschnitte von 3*3 Pixel, 5*5 Pixel oder generell N*N Pixel aufgeteilt. Es ist dabei vorteilhaft ungerade N zu wählen, da dann die Bildmitte durch ein Pixel definiert wird. Jeder Ausschnitt repräsentiert eine Teilmenge des Bildes und damit auch ein Teil der Kanten im Bild. Um diese Teilmengen als Fuzzy-Mengen zu benutzen muß jeder Pixel (Bildpunkt) fuzzifiziert werden, d.h jedem Pixel wird eine Zugehörigkeit zu seiner Teilmenge und damit auch zu einer Kante aus dem Intervall [0,1] zugewiesen. Dieses geschieht über eine Zugehörigkeitsfunktion [ti97], die die Übergänge in den Graustufen hervorhebt:

$$\mu_{Kanten}(g_{mn}) = 1 - \frac{1}{1 + \frac{1}{MAX} \sum |g_{mn} - g_{ij}|} \quad (3)$$

Es bedeuten MAX = Helligkeitsmaximum des Ausschnitts (lokal) oder des Bildes (global)

$\mu_{Kante}(g_{mn})$ = Zugehörigkeit,

g_{mn} = Grauwert der fuzzifiziert werden soll

g_{ij} = Eckpunkte des Ausschnitts

Es gibt dabei die Möglichkeit ein lokales oder ein globales Helligkeitsmaximum zu wählen.

Bei der Wahl eines lokalen Helligkeitsmaximums (Abbildung 2.6) werden die Kanten deutlicher hervorgehoben als bei einem globalen Helligkeitsmaximum (Abbildung 2.7), allerdings wird auch Rauschen verstärkt, falls es im Bild vorkommt.

Die Größe des Bildausschnitts sollte nicht zu groß gewählt werden (max 5*5 Pixel), da sonst lokale Eigenschaften verloren gehen können.

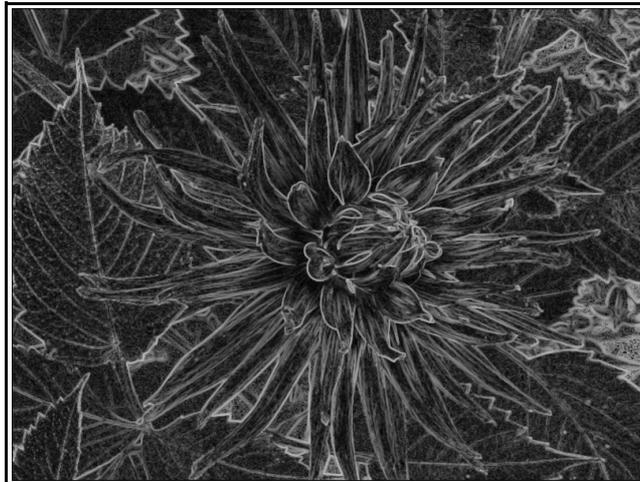


Abbildung 2.6 Kantenbild mit lokalem Maximum

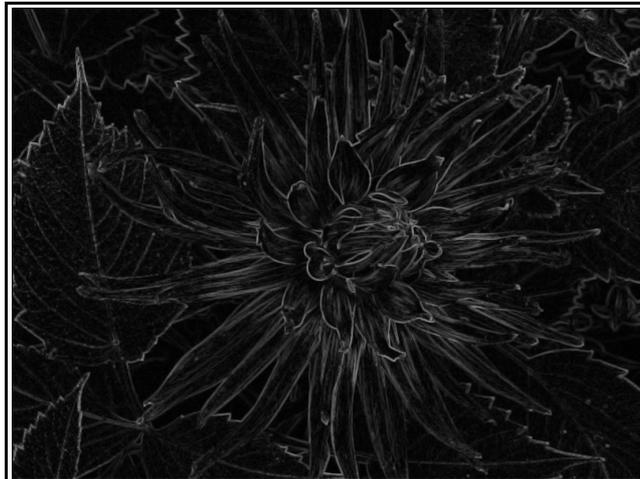


Abbildung 2.7 Kantenbild mit globalem Maximum

Für eine weitere Verstärkung der Kanten ist zusätzlich zur Ermittlung der Zugehörigkeit μ_i des Grauwertes g_i zu seiner Kante eine Modifikation $\mu \rightarrow \mu'$ nötig. Für diese Modifikation wird der folgende Intensitätsoperator benutzt, der die Übergänge (Kanten) im Bild noch zusätzlich hervorhebt (Abbildung 2.8).

$$\mu_i' = \begin{cases} 2 \cdot \mu_i^2 & \text{für } 0 < \mu_i \leq 0,5 \\ 1 - 2(1 - \mu_i)^2 & \text{für } 0,5 > \mu_i \geq 1 \end{cases} \quad (4)$$

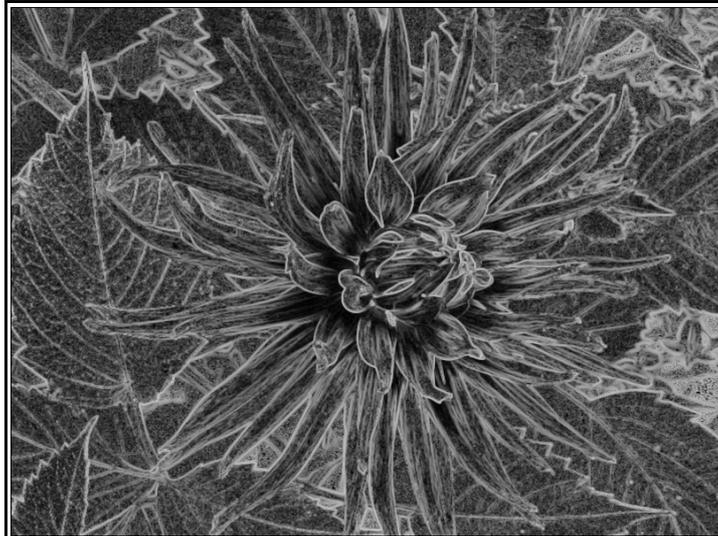


Abbildung 2.8 Kantenbild mit lokalem Maximum und Intensitätsoperator

Bei der Rücktransformation (Defuzzyfizierung), wird ein Zugehörigkeitswert μ_i' mit dem lokalen oder globalen Helligkeitsmaximum MAX des Ausschnitts multipliziert. Dies erzeugt einen Grauwert g_i' :

$$g_i' = MAX \cdot \mu_i' \quad (5)$$

Diese Rücktransformation erzeugt allerdings Negativ-Bilder (siehe Abbildungen 2.6 - 2.8), d.h. die hervorgehobenen Kanten werden heller dargestellt als das übrige Bild. Eine „normale“ Darstellung (Abbildung 2.9) erfolgt dadurch, daß g_i' von der höchsten Helligkeit (255 bei 8 Bit) abgezogen wird:

$$g_i^* = 255 - (MAX \cdot \mu_i') \quad (6)$$



Abbildung 2.9 "positiv" Darstellung eines Kantenbildes

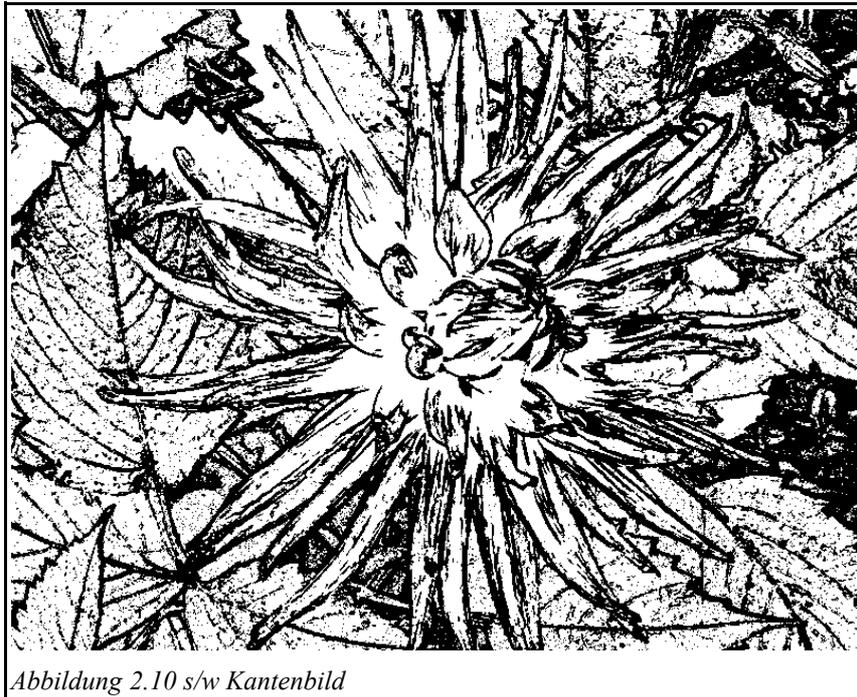
Es kann auch erwünscht sein, überhaupt keine Grauwerte mehr zu erhalten sondern nur schwarz/weiß Werte. Dann gilt:

$$g_i = \begin{cases} 0 & \text{falls } g_i < T \\ 255 & \text{falls } g_i > T \end{cases} \quad (7)$$

mit $0 \leq T \leq 255$

wobei Grauwerte $< T$ als schwarz und $> T$ als weiß dargestellt werden.

Ein Bild das mit diesem Merkmal defuzzyfiziert wird, besteht nur noch aus den Kanten seiner Objekte (Abbildung 2.10).



Ein Beispiel soll die Kantendetektion mit lokaler Fuzzyfizierung verdeutlichen:

Ein 3x3 Pixel Ausschnitt aus einem Graustufenbild hat die Eckpunkte mit den Grauwerten $g_{11}=195$, $g_{13}=100$, $g_{31}=95$ und $g_{33}=5$. Der Grauwert der fuzzyfiziert werden soll ist $g_{22}=45$.

195		100
	45	
95		5

Der lokale maximale Helligkeitwert $MAX = 195$. Mit dem Ausdruck (3)

$$\mu_{Kante}(45) = 1 - \frac{1}{1 + \frac{1}{195} \cdot (|195 - 45| + |100 - 45| + |95 - 45| + |5 - 45|)} \quad (8)$$

$$\Rightarrow \mu_{Kante}(45) \approx 0.6$$

Die Zugehörigkeit des Grauwertes $g_{22}=45$ zur Kante beträgt damit $\mu_{Kante}(45)=0,6$.

Die Defuzzyfizierung des Zugehörigkeitswert ist nun eine Multiplikation des Zugehörigkeitswert

mit dem lokalen Helligkeitsmaximum 195 (oder dem globalen des Bildes). Bei der Multiplikation mit dem lokalen Helligkeitsmaximum erhält man einen neuen Grauwert $g'_{22} = 117$. Dieser muss an die selbe Position in dem Bild eingefügt werden. Damit erhält man schließlich ein Kantenbild wie in Abbildung 2.6.

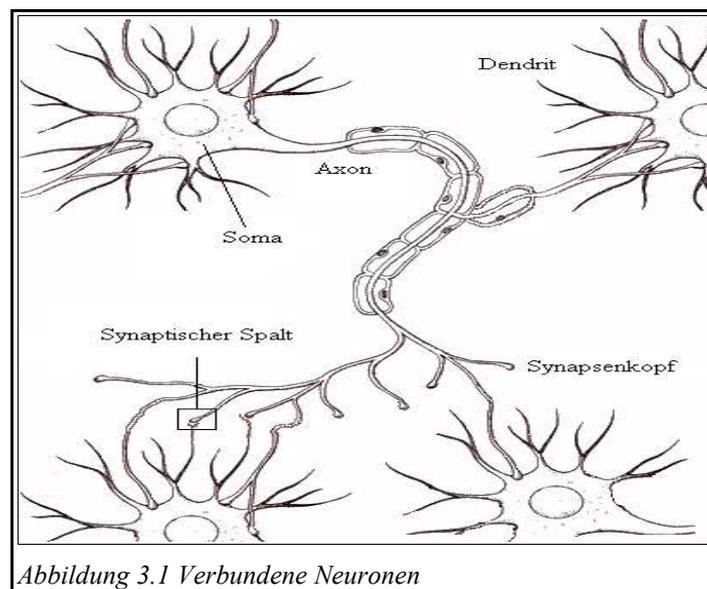
3 Das Visuale System

3.1 Einführung

Bei der Entwicklung Neuronaler Netze wird versucht die Lernfähigkeit intelligenter Lebewesen in technische Systeme abzubilden. Insbesondere die Gehirne höher Lebewesen gelten als anpassungsfähig und fehlertolerant und verblüffen durch ihre massiv parallele Organisation. Daher dient die Arbeitsweise des Gehirns der Neuroinformatik als Vorbild für künstliche neuronale Netze.

Neue Erkenntnisse im Aufbau und der Arbeitsweise des Gehirns haben zumeist zu neuen künstlichen neuronalen Netzmodellen geführt.

Der Mensch verfügt über eines der leistungsfähigsten Gehirne, die die Natur hervorgebracht hat. Es besteht aus ca. 10^{12} Zellen, davon sind ca. 10 % Neuronen (Nervenzellen).



Mit einer Anzahl von 40.000 Zellen pro mm^3 ist deren Dichte im Neokortex (Hirnrinde) am größten.

Die Funktion des Gehirns beruht auf dem Austausch von Informationen zwischen den Neuronen (Abbildung 3.1). Ein Neuron selbst besteht aus Soma (Zellkörper) und Axon (Reizleiter). Das Axon dient dabei zur Übertragung von Informationsimpulsen (Reize), die durch das Soma ausgelöst wurden, an nachgeschaltete Neuronen. Die Informationen werden dabei in eine Folge von Aktionspotentialen (AP) umgewandelt, d.h wenn eine Zelle gereizt wird, ändern sich an ihrer

Oberfläche (Membran) das Elektrische Potential (Spannung) und damit die elektrische Leitfähigkeit. Ist der Reiz stark genug, kommt es zu einem AP, das im Neuron einen Reiz erzeugt der dann an andere Neuronen über das Axon weitergeleitet wird. Im Anschluß an die Reaktion der Zelle baut sich dann wieder ein Ruhepotential auf.

Ein Neuron verfügt zur Abgabe von Reizen über Synapsenköpfe, die am Ende des Axons hängen und über einen synaptischen Spalt (Lücke zwischen Synapsenkopf und Zellmembran) zur Aufnahme von Reizen dienen. Ein ankommender Reiz (AP) veranlasst eine Ausschüttung von Neurotransmittern (Botenstoffe) im synaptischen Spalt, dadurch findet dann eine Umkodierung der Information statt.

Die Neurotransmitter summieren sich im Neuron und bei der Überschreitung eines Grenzwertes wird eine neue Information in ein Aktionspotential umgewandelt und losgeschickt.

Die Schaltzeiten einzelner Neuronen sind meist größer als 1 ms, trotzdem ist die Leistungsfähigkeit des Gehirns, insbesondere bei der Lern- und Abstraktionsfähigkeit, aber auch bei der Fehlertoleranz jedem herkömmlichen Computer überlegen. Dieses liegt nicht zuletzt an der hohen Parallelität der Neuronen. Auch der Ausfall einzelner Neuronen schadet der Gesamtfunktion des Gehirns meist nur unwesentlich. Diese herausragende Eigenschaft ist im wesentlichen der Grund für die gegenwärtigen Anstrengungen, technische System nach dem Vorbild der Funktionsweise des Gehirns zu verwirklichen.

3.2 Grundlagen

Die Menschen, aber auch viele Tiere, verlassen sich am meisten auf das Sehen beim täglichen Überlebenskampf. Die meisten Informationen werden über das Sehen aufgenommen, deswegen stellt es auch für die meisten Menschen den wichtigsten der fünf Sinne dar. Denn unsere Augen sind der Schlüssel für unsere moderne Welt, die überwiegend auf das Sehen ausgerichtet ist. Eine Nachbildung des „visuellen Systems“ als ein Neuronales Netz ist deswegen eine besondere Herausforderung.

3.2.1 Die Retina

Im Auge wird das Licht, das von Objekten erzeugt oder reflektiert wird, durch Hornhaut und Linse gebündelt und erzeugt auf der Retina (Netzhaut) ein Bild dieses Objekts (Abbildung 3.2).

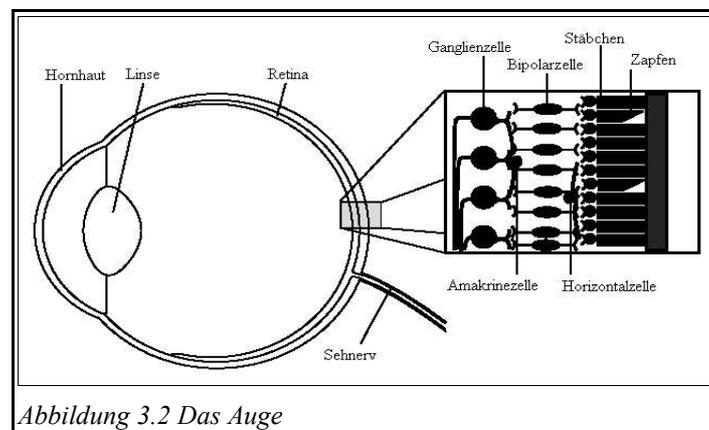
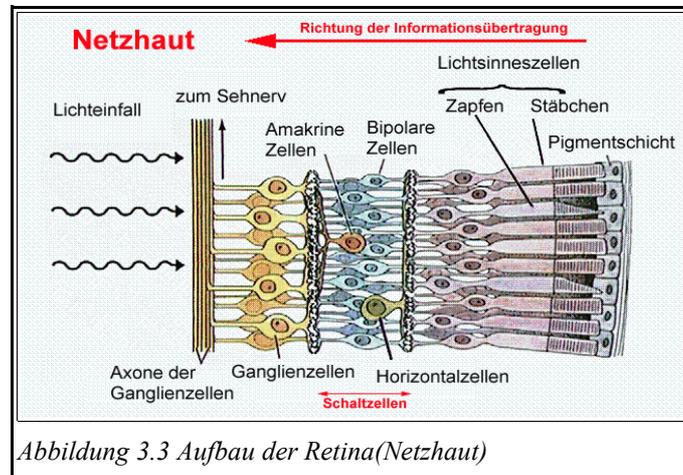


Abbildung 3.2 Das Auge

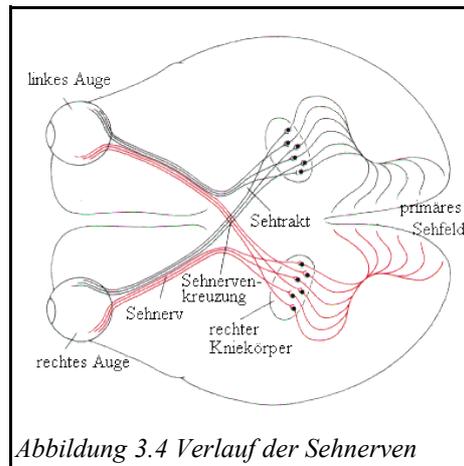
Die Retina (Abbildung 3.3) selbst besteht aus drei Schichten von Neuronen und bildet den Hintergrund des Auges. Bei Licht werden die Photorezeptoren (Zapfen und Stäbchen für das Farb- und Helligkeitssehen) gereizt, welche die letzte Schicht der Retina bilden, d.h. das Licht muß erst andere Zellschichten durchdringen, um die Photorezeptoren zu erreichen. Das ist bei allen Linsenaugen der Fall und wird „inverses Auge“ genannt. Die mittlere Schicht besteht überwiegend aus Bipolarzellen, welche die Photorezeptoren mit den Ganglienzellen verbinden. Zu einem kleinen Teil besteht die mittlere Schicht auch aus Amakrinezellen und Horizontalzellen, wobei die Horizontalzellen die Photorezeptoren sowie die Amakrinezellen die Ganglienzellen untereinander verschalten. Die erste Schicht besteht aus Ganglienzellen. Diese leiten die von den Photorezeptoren

übermittelten Informationen über ihre Axone (Reizleitung der Nerven) an den Sehnerv weiter.



Ein großer Anteil der Informationen, welche das Auge empfängt, wird schon in der Retina aufbereitet. Ein Auge besitzt ca. 125 Millionen Stäbchen und 5,5 Millionen Zapfen, aber nur 900.000 Nervenfasern verbinden das Auge mit dem Gehirn. Das liegt daran, dass nicht jedes Stäbchen oder jeder Zapfen mit einer Ganglienzelle verbunden ist. Vielmehr gibt es Bereiche in denen viele Stäbchen und Zapfen mit einer Ganglienzelle verbunden sind, aber auch Bereiche in der zentralen Region des Auges (Fovea centralis, Sehgrube), wo jeder Photorezeptor mit einer Ganglienzelle verbunden ist. Dieser zentrale Bereich (Sehgrube) ermöglicht das beste Sehen, während die anderen Bereiche nur ein vages Bild der Umgebung vermitteln können.

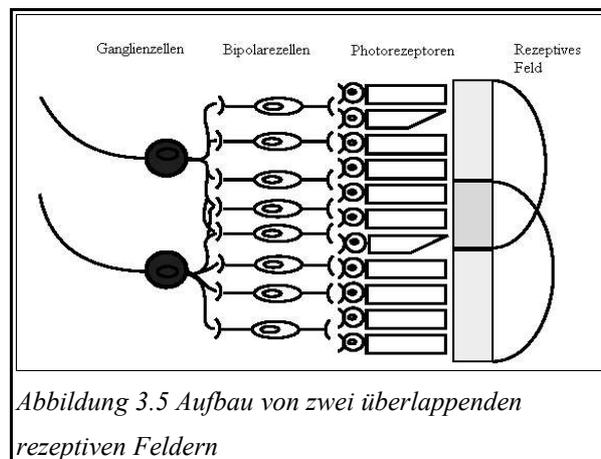
Der Mensch verfügt über zwei Augen für eine dreidimensionale Sicht der Umgebung. Von jedem Auge kommt ein Sehnerv (Abbildung 3.4), der sich mit dem Sehnerv des anderen Auges einmal kreuzt (Sehnervenkreuzung), wobei eine Hälfte eines Sehnervs die Seite wechselt und die andere Hälfte auf der Seite des zugehörigen Auges bleibt. Nach dem Kreuzen enthält jeder Sehtrakt auch die Hälfte der Informationen des anderen Auges. Dieses dient zur Überlagerung der Sehfelder und ermöglicht damit das dreidimensionale Sehen. Der Sehtrakt endet im linken oder rechten Sehzentrum und wird dann weitergeführt bis zum kortikalen Sehzentrum.



3.2.2 Das rezeptive Feld

Sehen ist mehr als nur der Transport von Informationen vom Auge ins Gehirn. Hätte die Natur nur dies bezweckt, hätten Nervenstränge von jedem Auge zum Sehzentrum genügt. Aber es geschieht wesentlich mehr als nur der Transport, denn die Zellen im kortikalen Sehzentrum können auf sehr spezielle Muster reagieren, z.B. auf Linien einer bestimmten Länge, Bewegungen von Linien in bestimmte Richtungen usw.

Im Bereich außerhalb der Sehgrube sind benachbarte Photorezeptoren über mehrere Bipolarzellen mit nur einer Ganglienzelle verbunden, eine solche Gruppe wird „rezeptives Feld“ genannt und hat folgenden Aufbau (Abbildung 3.5).



Es gibt allerdings Überlappungen von Photorezeptoren, die dann zu mehreren benachbarten rezeptiven Feldern gehören können. Es existieren zwei unterschiedliche Typen von Bipolarzellen:

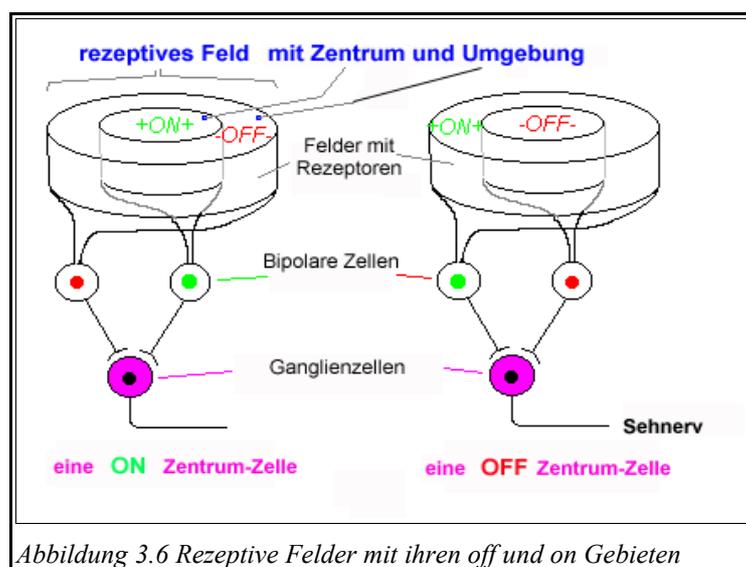
Die eine Sorte reagiert auf eine Anregung durch die Photorezeptoren mit einer Erregung (Hyperpolarisierung) und leitet diesen Reiz dann an ihre Ganglienzelle weiter, sie gehört zum „ON-Typ“. Die andere Bipolarzellsorte reagiert genau umgekehrt: Sie wird durch eine ankommende Erregung sozusagen abgeschaltet (depolarisiert) und leitet damit keinen Reiz an ihre Ganglienzelle. Sie gehört zum „OFF-Typ“.

Diese beiden unterschiedlichen Typen von Bipolarzellen bilden zwei unterschiedlich strukturierte Typen von rezeptiven Feldern:

- i) ON-Zentrum und OFF-Umgebung
- ii) OFF-Zentrum mit ON-Umgebung.

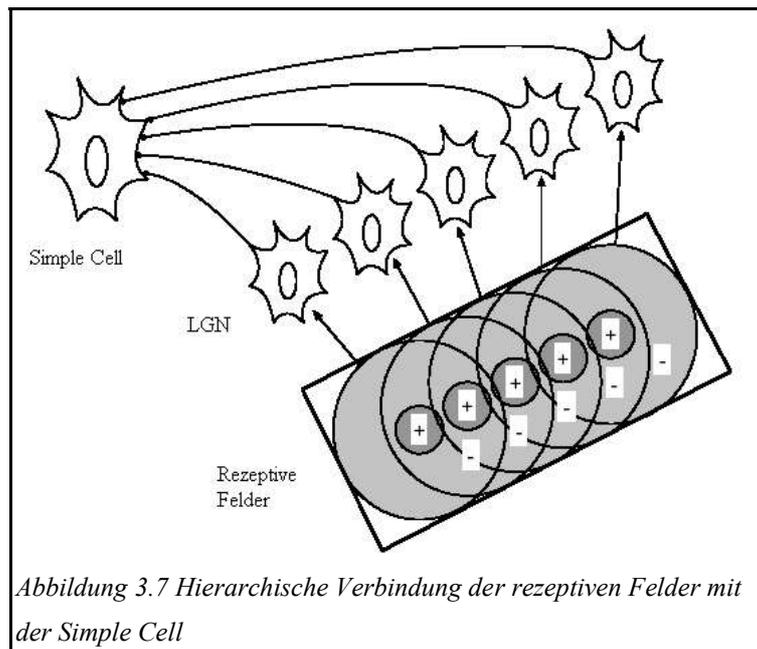
Das Zentrum eines rezeptiven Feldes verhält sich dabei immer entgegengesetzt (antagonistisch) zu seiner Umgebung. Bei beiden Feldtypen wird die Information an eine Ganglienzelle mit jeweils entgegengesetzter Funktionsweise weitergeleitet, die „ON-Zentrums-Ganglienzelle“ und die „OFF-Zentrums-Ganglienzelle (Abbildung 3.6)“.

Die Größe der rezeptiven Felder ist in der Sehgrube am kleinsten und vergrößert sich kontinuierlich in Richtung des Randes.



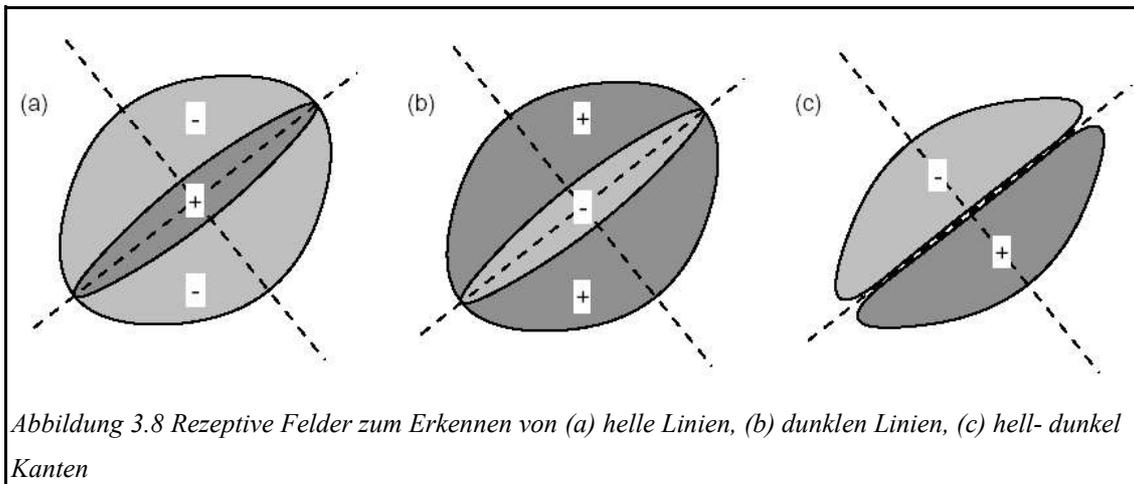
3.2.3 Die Simple Cell

Ein rezeptives Feld bzw. mehrere rezeptive Felder sind über das „Lateral Geniculate Nucleus“ (LGN) mit nur einer Zelle (Neuron) im kortikalen Sehzentrum verbunden. Diese wird „Simple Cell“ genannt (Abbildung 3.7).



Es existieren insgesamt drei unterschiedliche Typen von rezeptiven Feldern die in den Abbildungen 3.8a, 3.8b und 3.8c dargestellt werden.

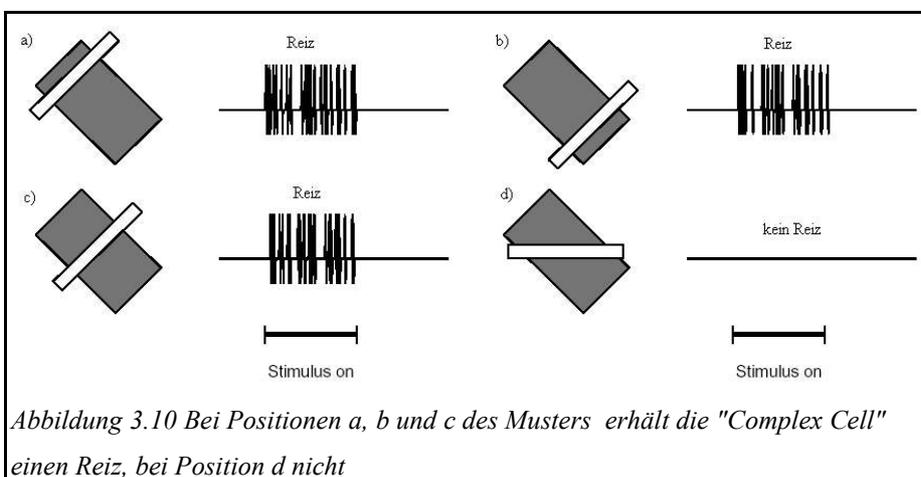
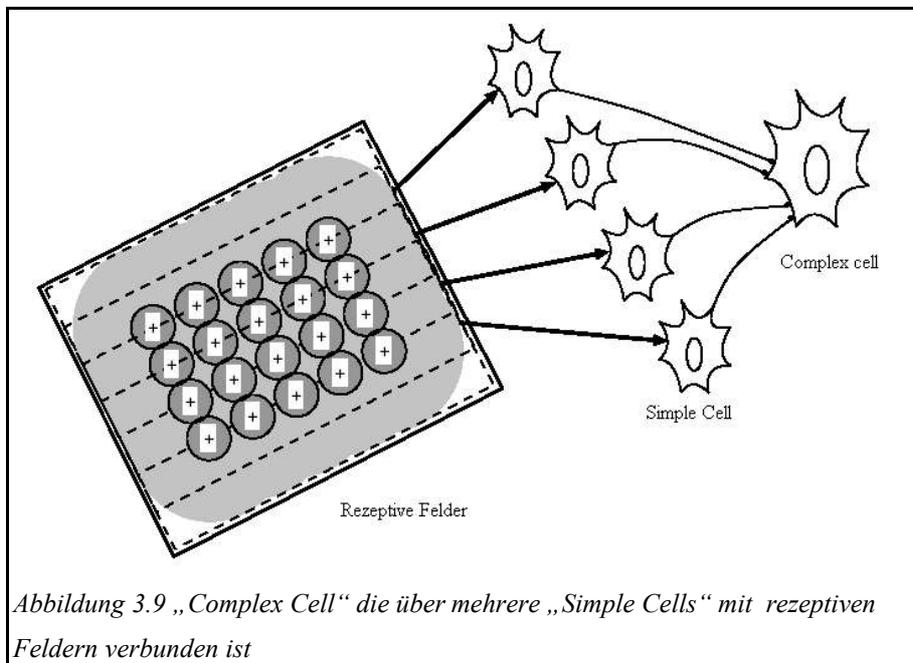
Um eine Reaktion in der „Simple Cell“ auszulösen, muß der Teil des rezeptiven Feldes mit den „anregenden“ (+) Photorezeptoren beleuchtet sein, während die „hemmenden“ (-) Photorezeptoren im dunkeln liegen müssen. Das bedeutet, daß das rezeptive Feld (a) eine helle Linie auf dunklem Hintergrund erkennt, wenn diese auf den (+)-Bereich fällt. Das rezeptive Feld (b) verhält sich genau entgegengesetzt. Es erkennt eine dunkle Linie auf hellem Hintergrund. Das rezeptive Feld (c) schließlich erkennt einen Übergang von hell (+) nach dunkel (-), d.h. eine Kante.



Die drei rezeptiven Felder sind geometrisch so aufgebaut, daß sie sehr gut zur Erkennung von einfachen visuellen Mustern wie Linien oder Kanten in einer bestimmten Ausrichtung dienen. Es hat sich gezeigt, daß „Simple Cells“ besonders gut reagieren, wenn die Muster in ihrer geometrischen Ausrichtung mit denen der „hemmenden (-)“ und „anregenden (+)“ Bereiche der rezeptiven Felder übereinstimmen. Eine Verschiebung einer Linie oder Rotation von nur 10° zu jeder Seite reduziert die Reaktion der „Simple Cell“ dramatisch. Es existieren deswegen rezeptive Felder in beliebiger Orientierung zur Erkennung von allen möglichen Ausrichtungen der Muster.

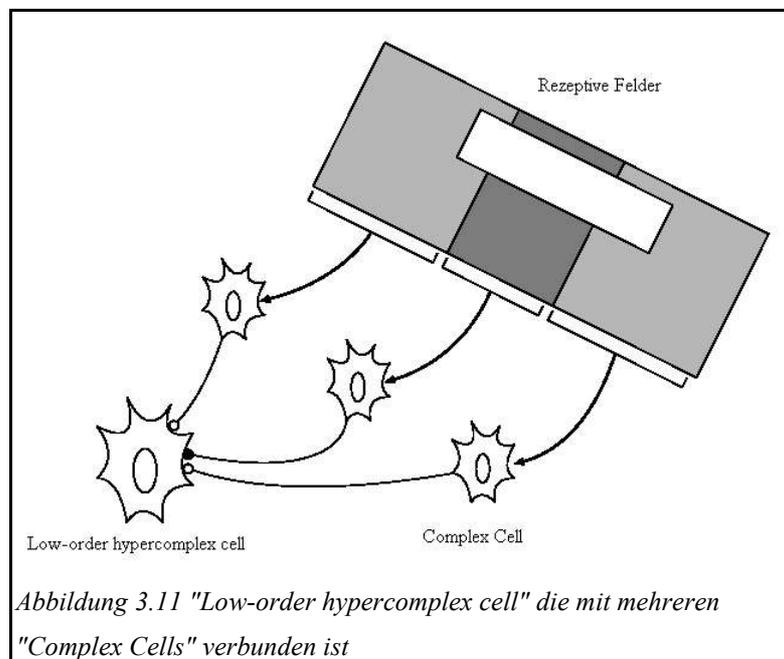
3.3 Die Complex Cell

Eine „Complex Cell“ (Abbildung 3.9) ist mit mehreren „Simple Cells“ verbunden. Diese „Simple Cells“ sind nur mit solchen rezeptiven Feldern verbunden, die Muster mit derselben Ausrichtung erkennen. Die „Complex Cell“ reagiert besser auf bewegte als auf stationäre Muster. Wie ihr Gegenstück, die „Simple Cell“ reagiert auch die „Complex Cell“ auf Linien und Kanten über die rezeptiven Felder der „Simple Cells“. Für die „Complex Cell“ ist aber die räumliche Position des Musters irrelevant, solange das Muster korrekt auf das rezeptive Feld ausgerichtet ist. In Abbildung 3.10 sind vier Muster (a-d) in unterschiedlichen Positionen dargestellt. Nur das Muster „d“, welches nicht nur die Position, sondern auch die Ausrichtung verändert hat, wird nicht erkannt.



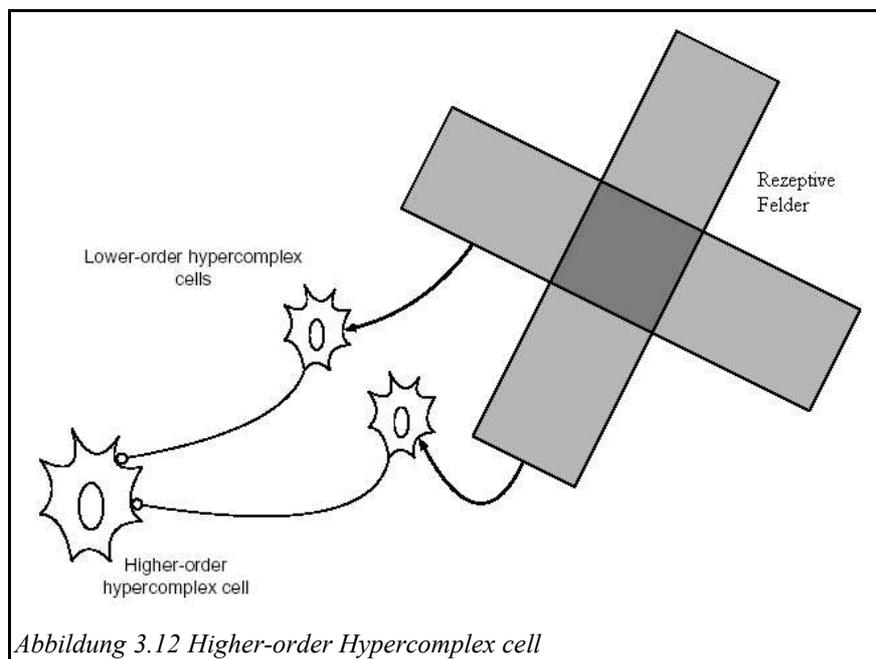
3.3.1 Die Lower-order hypercomplex Cell

Sowohl „Simple Cells“ als auch „Complex Cells“ reagieren, wenn Muster richtig ausgerichtet sind. Dies gilt auch, wenn die Länge der Muster (Linie oder Kante) die Ausdehnung des rezeptiven Feldes übersteigt. Eine Besonderheit der „Lower-order hypercomplex cell“ ist, daß bei ihr nur bei einer bestimmten Länge des Musters eine maximale Reaktion hervorgerufen wird. Dieses Verhalten wird „end-stopping“ genannt und kann mit „anregenden“ und „hemmenden“ Regionen erklärt werden. Eine „Lower-order hypercomplex cell“ erhält Information über „Complex Cells“ von verschiedenen nebeneinanderliegenden rezeptiven Feldern. Ein einfaches Beispiel ist in Abbildung 3.11 dargestellt. Die „Lower-order hypercomplex cell“ reagiert maximal, wenn das Muster genau auf das mittlere rezeptive Feld passt. Ist das Muster kleiner, so verringert sich auch die Reaktion der „Lower-order hypercomplex cell“. Übersteigt die Größe des Musters das rezeptive Feld und bedeckt auch Teile der angrenzenden rezeptiven Felder, so verringert sich auch die Reaktion in der „Lower-order hypercomplex cell“ und zwar um so mehr, je größer die Bedeckung der angrenzenden rezeptiven Feldern ist. Auf Grund dieses Mechanismus ist es möglich, die Größe von Mustern zu erkennen.



3.3.2 Die Higher-order hypercomplex Cell

Die „Higher-order hypercomplex cell“ ist mit zwei „Lower-order hypercomplex cells“ verbunden. Das ermöglicht mit dem im vorherigen Abschnitt beschriebenen Mechanismus auch die Identifizierung von nach zwei Dimensionen (x und y) ausgerichteten Mustern (Ecken oder Kurven) (Abbildung 3.12).



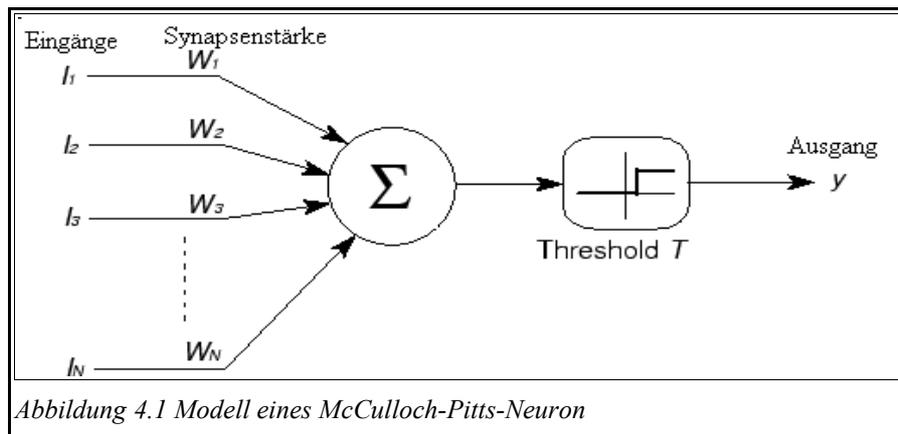
3.4 Nachtrag

Den Aufbau und die Funktionsweise dieses „Visuellen Systems“ entdeckten David H. Hubel und Torsten N. Wiesel um 1960 [huwi59]. Sie beschrieben den Aufbau dabei als linear und streng hierarchisch: Photorezeptoren sind über die Ganglienzellen mit dem Lateral Geniculate Nucleus verbunden und weiter über die Simple-, Complex-, zu den hypercomplex cells (lower- und higher order). Diese Arbeit wurde im Jahre 1981 mit dem Nobelpreis für Medizin geehrt. Im Jahr 1975 bemerkte J. Anthony Movshon [mo75], daß einige Verhaltensweisen nicht mit einem einfachen hierarchischen Aufbau beschrieben werden konnten. In weiteren Experimenten von Hammond, MacKay [hama75] und Sillito [si75] bestätigte sich dieses. Dies lässt die Annahme zu, daß das „visuelle System“ wahrscheinlich noch wesentlich komplexer ist als oben beschrieben. Vermutlich werden Informationen nicht nur seriell (hierarchisch) sondern auch parallel verarbeitet.

4 Aufbau und Arbeitsweise des Neocognitrons

4.1 Einführung

Im Jahr 1943 entwickelten die Mathematiker McCulloch und Pitts [mcp43] ein erstes Modell (Abbildung 4.1), das die wesentlichen Eigenschaften einer Nervenzelle erfassen sollte.



Ein McCulloch-Pitts-Neuron erzeugt einen Ausgangswert (es „feuert“) $y > 0$ genau dann, wenn die Eingangswerte I_i mit der Synapsenstärke (Gewicht) w_i multipliziert werden und die Summe der Produkte den Schwellwert T (Threshold) überschreitet.

$$y = \sum_{i=1}^n I_i \cdot w_i - T \quad (9)$$

Ursprünglich war der Schwellwert T ein Wert aus dem Bereich $[0;1]$ oder $[-1;1]$.

Später wurde der Bereich von T auf den ganzen Bereich der reellen Zahlen ausgedehnt.

Damit das Neuron sich anpassen kann und auf Eingangssignale unterschiedlich reagieren kann (etwas erlernen kann) ist es notwendig, die Synapsenstärke w_i zu verändern. Die Veränderung Δw_i wird Hebbische Lernregel genannt:

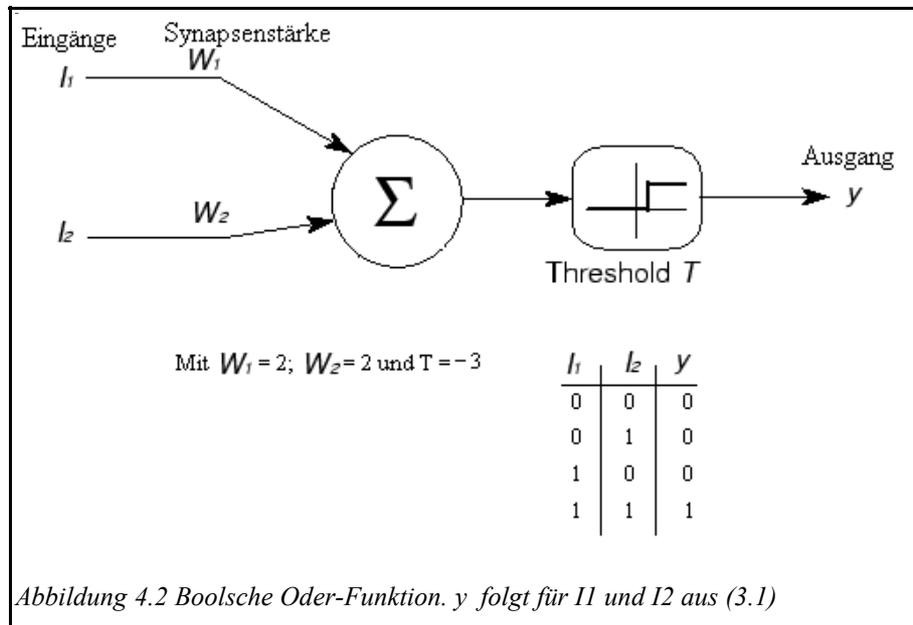
$$\Delta w_i = \varepsilon \cdot y \cdot I_i \quad (10)$$

es folgt dann :

$$w_{i(\text{neu})} = \Delta w_i + w_{i(\text{alt})} \quad (11)$$

Der Parameter $\varepsilon > 0$ wird dabei als Lernschrittkonstante bezeichnet.

Das Beispiel in Abbildung 4.2 zeigt ein Neuron, das die Funktionsweise einer „Oder-Funktion“ schon „erlernt“ hat.



Dabei können die Eingangswerte I_1 , I_2 und der Ausgangswert y den Wert 0 oder 1 annehmen und bilden mit dem Neuron eine „Oder-Funktion“ nach.

Eines der ersten Netzmodelle wurde 1958 von Rosenblatt [ro58]] entwickelt. Dieses Modell, „Perceptron“ genannt, besteht aus einer Schicht parallel arbeitender McCulloch-Pitts-Neuronen. Es stellt einen der Meilensteine in der Geschichte der Neuroinformatik dar.

Die meisten heutigen Modelle basieren auf dem McCulloch-Pitts-Neuron oder einer Variante. Auch das Cognitron und sein Nachfolger, das Neocognitron, basieren auf dieser Grundidee, wenn auch in stark erweiterter und wesentlich komplexerer Form.

4.2 Das Cognitron

In der ersten Hälfte der siebziger Jahre entwickelte Kunihiko Fukushima das Cognitron [fu75]. Das Cognitron war der direkte Versuch, das „Visuelle System“ von Säugetieren nachzuahmen. Fukushima nutze dabei sowohl die Ergebnisse von Hubel und Wiesel, als auch die von Blakemore und Cooper [blco70]. Blakemore und Cooper zeigten, daß die Neuronen, die zur Mustererkennung im „Visuellen System“ dienen, sich in Säugetieren so anpassen, daß sie nur Muster erkennen, die auch wirklich in deren Umgebung vorkommen. Diese Idee war nicht neu. Denn schon sechszwanzig Jahre vorher stellte Hebb diese Hypothese auf. Fukushima erweiterte dieses Konzept auf ein mehrschichtiges Neuronales Netz das sein „erlerntes“ Wissen in jeder Schicht speichert.

Die Hebbsche Lernregel schreibt vor, daß nur eine Verstärkung des Gewichtes (Synapsenstärke) auftritt, wenn sowohl der Eingang als auch das Neuron, d.h der Ausgang selbst, aktiv sind. Die in den siebziger Jahren entstanden Neuronalen Netze wenden diese Regel an. Die Neuronen in diesem, auf dem Perceptron basierendem Modell, waren praktisch alle gleich miteinander verbunden. Solch ein Netz konnte aber noch nicht als „selbst-organisierend“ angesehen werden. Bei einem „selbst-organisierenden“ Neuronalen Netz beeinflussen die in der „Lernphase“ aktivierten Neuronen auch benachbarte Neuronen und verändern diese mit.

Fukushima erweiterte die Hebbsche Lernregel. Er nutze dabei, daß im „Visuellen System“ benachbarte Neuronen ihre Information von vorhergehenden Zellen der gleichen Region erhalten. Um zu verhindern, daß sich benachbarte Neuronen gleich verhalten, stellte er folgende These auf:

„Die Verbindung vom Neuron x zum Neuron z wird nur verstärkt, wenn diese beiden Bedingungen gelten“ :

1. das erste Neuron x „feuert“,
2. keines der benachbarten Neuronen des Neurons z „feuert“ stärker als das Neuron z selbst.

Dies bedeutet folgendes: Wenn zwei Neuronen x und z hintereinander geschaltet sind, wird die Synapsenstärke w_i von beiden Neuronen nur dann erhöht, wenn das Neuron x einen Ausgabewert y

liefert und der Ausgabewert y von Neuron z der größte in einer bestimmten, nicht näher definierten Nachbarschaft ist.

Dieses Prinzip der „Selbst-Organisation“ wurde erfolgreich mit dem Cognitron getestet. Das Cognitron bewies dabei seine Fähigkeit, verschiedene zweidimensionale Muster zu „erlernen“: Spezielle Neuronen im Cognitron reagierten nach einigen Lerndurchgängen nur noch auf bestimmte Muster in der Eingabe. Dieses führte zu einer Spezialisierung von einzelnen Neuronen, wie auch im Vorbild, dem „Visuellen System“.

4.3 Das Neocognitron

Das Cognitron verwendet ähnliche Verhaltensweisen wie das Gehirn. Es besteht aus mehreren Schichten (Layer) von Neuronen, wobei jedes Neuron nur die Information von gleichen Regionen der vorhergehenden Schicht erhält. Die Neuronen des Cognitrons selbst können so trainiert werden, daß sie auf unterschiedliche Muster unterschiedlich reagieren. Allerdings ist es nicht in der Lage, angelernte Muster die verschoben oder leicht verändert (deformiert) werden, wieder zu erkennen. Fukushima erweiterte deswegen das Cognitron um einige von Hubel und Wiesel behandelte Aspekte und entwickelte das Neocognitron. Das Neocognitron besitzt die Fähigkeiten des Cognitron, kann aber darüber hinaus verschobene und deformierte Muster erkennen. Damit ist das Neocognitron das erste Neuronale Netz, welches das menschliche „Visuelle System“ (in diesem Umfang) nachahmt.

Fukushima änderte mit der Zeit den Anwendungszweck des Neocognitron von einem künstlichen „Visuellen Systems“ zu einem System zur Handschriftenerkennung.

Der allgemeine Aufbau und die Arbeitsweise des Neocognitron werden in den nächsten Kapiteln anhand des Beispiels der Erkennung der Zahlen 0 bis 9 näher erklärt.

4.4 Die Eingangsinformationen des Neocognitrons

Obwohl wir die Umwelt dreidimensional wahrnehmen, kann die Retina nur zweidimensionale Eindrücke vermitteln. Erst durch Überlagerung von Bildern unserer beiden Retina entsteht ein dreidimensionales Bild. Das menschliche „Visuelle System“ ist perfekt darauf abgestimmt, Informationen von Millionen von Photorezeptoren zu verarbeiten. Das Neocognitron ist eine Simulation dieses „Visuellen System“, allerdings sind Computer, die Daten seriell verarbeiten, mit einer solchen Fülle von Informationen restlos überfordert. Zur Eingabe der Muster und zur weiteren Bearbeitung im Neocognitron werden deswegen meist rechteckige Pixelfelder benutzt, die als grob gerastertes Binärbild (Schwarz/Weiß) einer Retina angesehen werden können. Zu Beginn der achtziger Jahre nutzte Fukushima 20 x 20 Pixel große Felder für seine Handschriftenerkennung. Mit der Weiterentwicklung der Computer können in der Zwischenzeit aber größere Felder genutzt werden.

Das Neocognitron soll ein Modell des „Visuellen Systems“ darstellen. Es besitzt aber keinen Mechanismus zur Aufbereitung von Bildern, deswegen ist eine Kantendetektion (siehe Kapitel 2) nötig, um die Bilder sinnvoll mit dem Neocognitron zu verarbeiten.

4.5 Erster Layer simple cells

Die Terminologie „erster Layer simple cells“ beschreibt den ersten Verarbeitungsschritt eines Neocognitrons und macht deutlich, wie stark Fukushima von der Arbeit von Hubel und Wiesel beeinflusst wurde. Es existieren zwar einige Unterschiede zwischen der ersten Phase der Verarbeitung im „Visuellen System“ von Säugetieren und dem ersten Layer des Neocognitrons, aber die wesentlichen Arbeitsweisen sind identisch. Denn beide Systeme extrahieren einfache „Merkmale“ von der „Visuellen Eingabe“ und verarbeiten sie weiter. Der Begriff „Merkmal“ bedeutet in diesem Zusammenhang ein kleiner Ausschnitt oder Teil eines Musters, z.B. eine Linie oder Kante.

Der Mechanismus zur Erkennung von Mustern beim Neocognitron basiert auf der Idee, die auch das „Visuelle System“ von Säugetieren nutzt: Muster werden in „Merkmale“ (Linien oder Kanten) zerlegt. Eine Erkennung erfolgt dann, wenn eine bestimmte Anordnung der „Merkmale“ einem erlernten Muster zugeordnet werden kann.

Fukushimas Neocognitron benutzt bei der Handschriftenerkennung der Zahlen 0 bis 9 genau diesen

Mechanismus. Es werden die Eingangsinformation, d.h. eine Zahl zwischen 0 und 9 in ihre „Merkmale“ zerlegt. In den höheren Layern (Schichten) werden diese Merkmale wieder Stück für Stück zusammengesetzt und verglichen, ob sie am Ende einer der angelernten Zahlen zugewiesen werden können. Im ersten Layer des Neocognitron, das hier zur Erklärung verwendet wird, werden die Zahlen in sehr einfache „Merkmale“ der Größe 3x3 Pixel zerlegt.

Ein Unterschied zwischen dem Neocognitron und dem „Visuellen System“ von Säugetieren besteht in der Organisation der Neuronen. Fukushima benutzte eine ganz bestimmte Konvention für den Aufbau des Neocognitrons. Fukushima erschuf auch eine neue Bezeichnung für die Komponenten aus denen das Neocognitron besteht, um eine Abgrenzung zum „Visuellen System“ der Säugetiere zu schaffen.

S-cell → Simple Cell

C-cell → Complex Cell

S-planes → Gruppe oder Fläche von Simple Cells, die alle auf das selbe „Merkmal“ reagieren

C-planes → Gruppe oder Fläche von Complex Cells, die alle auf das selbe „Merkmal“ reagieren

Der erste Layer besteht zunächst aus einer Gruppe oder Fläche von Simple Cells (S-planes), gefolgt von einer Gruppe von Complex Cells (C-planes). Zusätzlich führte Fukushima die V-cell ein, für die es kein Analogon im „Visuellen System“ von Säugetieren gibt. Die Aufgabe der V-cell ist es, die Reaktion der Simple Cells bei der Erkennung von „Merkmalen“ abzuschwächen, die nicht zur einer der schon erlernten Zahlen 0 bis 9 gehören. Für das Verständnis des Aufbaus des Neocognitrons ist die V-cell ohne Belang. Sie wird deswegen erst später genauer erklärt.

In Abbildung 4.3 wird der erste Layer des Neocognitron zur Erkennung der Zahlen 0 bis 9 mit zwölf S-planes dargestellt. Zunächst wird nicht darauf eingegangen, wie die 3x3 Pixel großen „Merkmale“ aus der Zahl 9 (UC0) genau ausgewählt werden eine Erklärung folgt im Abschnitt über das Anlernen des Neocognitrons. Das Eingangsmuster ist die Zahl 9, dargestellt in 19x19 Pixel. Zwischen dem Eingangsmuster UC0 und jedem der zwölf S-planes (US1.0 bis US1.11) sind die „Merkmale“ als 3x3 Pixel Bild dargestellt. Jede S-Plane besteht dabei aus einer bestimmten Anzahl von S-cells (Gruppe), alle S-cells in dieser Gruppe reagieren auf das gleiche „Merkmal“. Die Stärke der Reaktion einer S-cell wird mit verschieden großen schwarz ausgefüllten Quadraten in jedem S-plane (US1.0 bis US1.11) dargestellt. Um so größer die Reaktion einer S-cell um so größer ist das Quadrat.

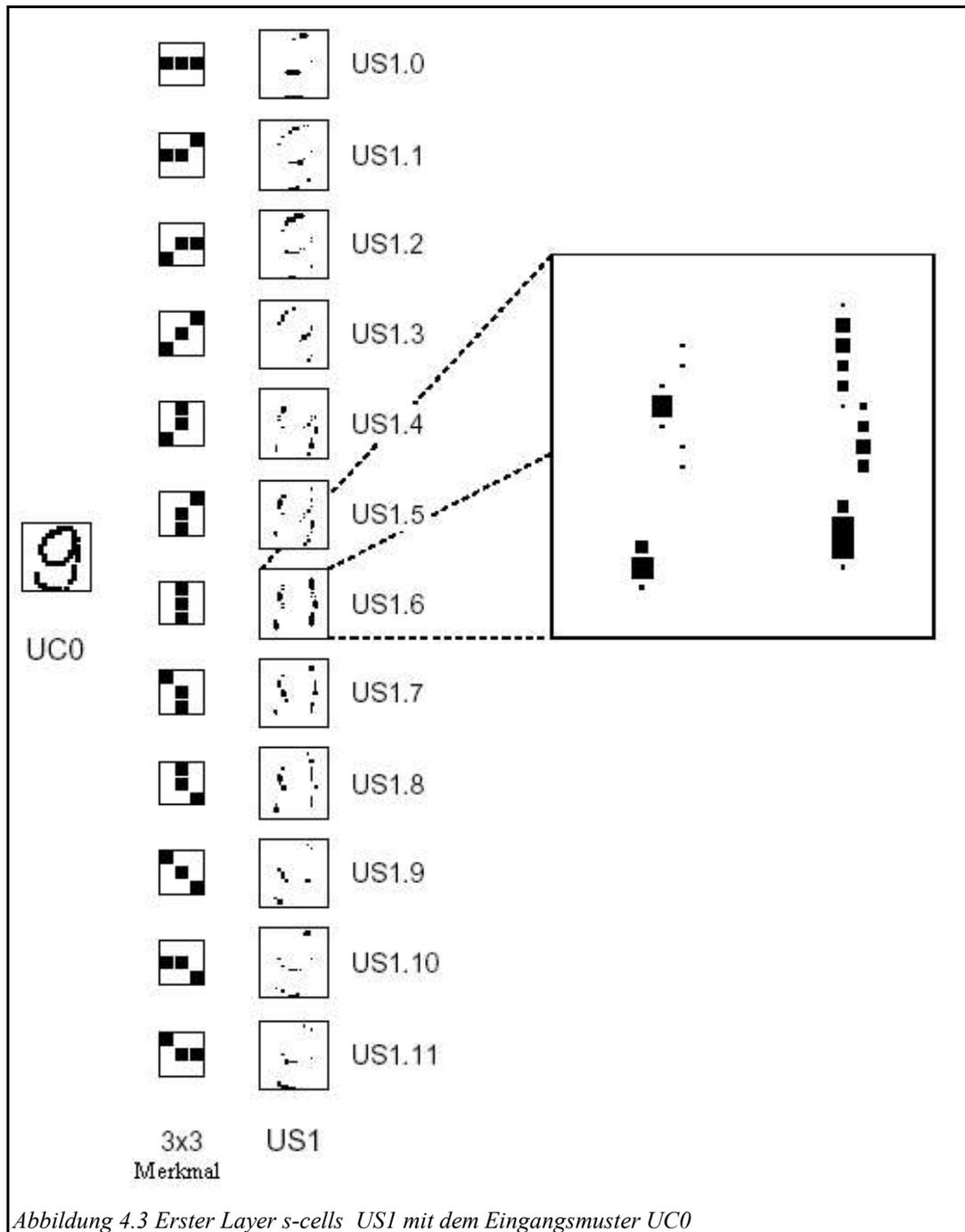
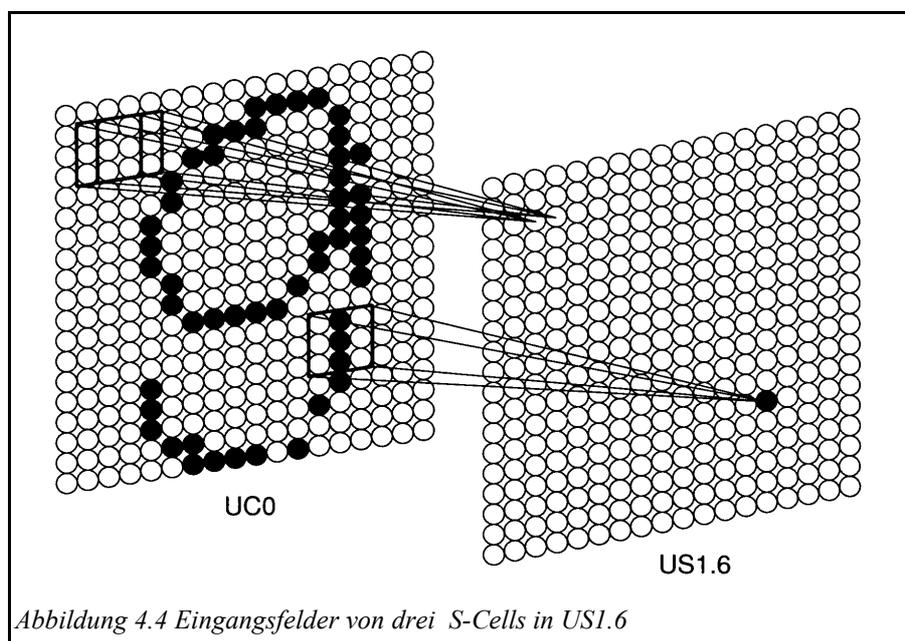


Abbildung 4.3 Erster Layer s-cells US1 mit dem Eingangsmuster UC0

Abbildung 4.3 zeigt auch eine vergrößerte Darstellung des S-plane US1.6. In dieser Größe ist erkennbar, daß die größte Reaktion der S-cells an den Stellen entsteht, wo das „Merkmal“ von US1.6 (vertikale Linie) genau mit dem Eingangsmuster (UC0) übereinstimmt. Bei nicht kompletter Übereinstimmung, z.B. wenn von dem „Merkmal“ der vertikalen Linie von US1.6 nur zwei oder ein Pixel mit dem Eingangsmuster übereinstimmen, entstehen auch nur kleine Quadrate in US1.6. Jede S-plane kann dabei als eine Art zweidimensionale Korrelation zwischen dem Eingangsmuster

und dem 3x3 Pixel großen „Merkmal“ angesehen werden.

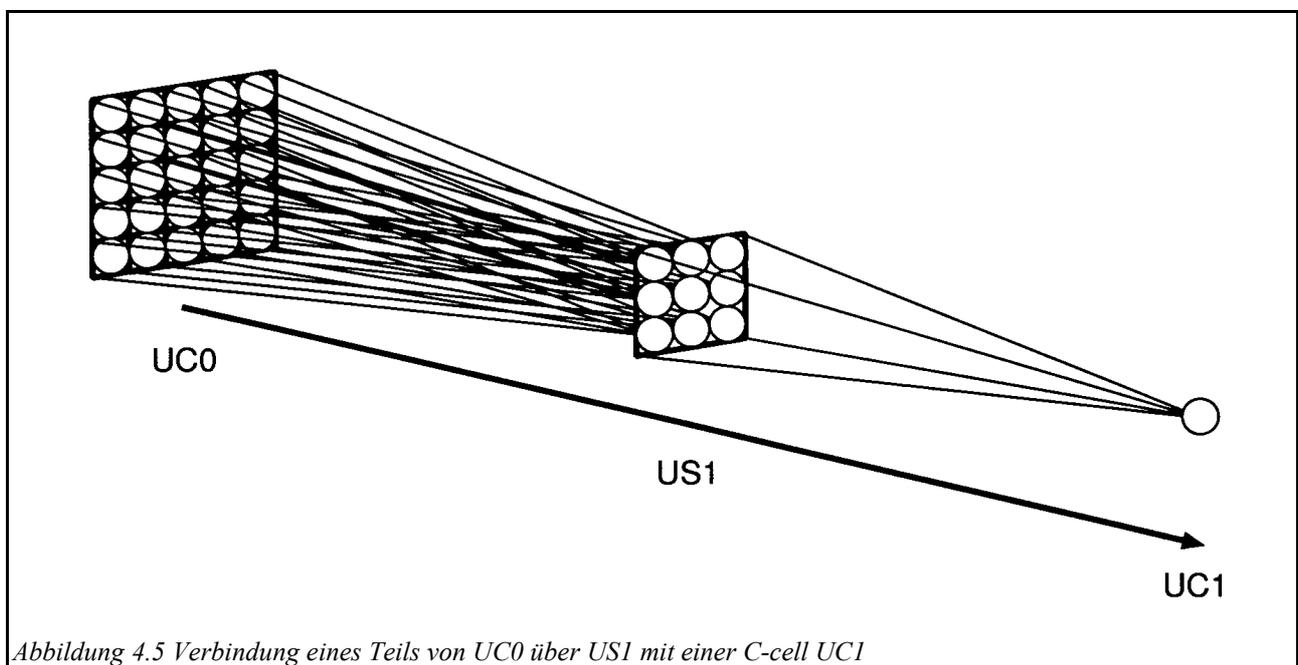
Jede S-cell im S-plane US1.6 (wie jeder andere S-plane des ersten Layer) erhält seine Eingangsinformationen von einem 3x3 Pixel großen Feld von dem vorherigen Layer. In diesem Fall, also von UC0. Entsprechend den rezeptiven Feldern in Abbildung 2.4 des vorherigen Kapitels besitzen auch die S-cells überlappende Felder. Dabei „besitzt“ jede S-Cell ein eigenes Eingabefeld und deckt dabei den gesamten vorherigen Plane (hier UC0) durch Überlappung ab. Es ist eine Art des Abrasterns des gesamten Feldes UC0 (Abbildung 4.4) .



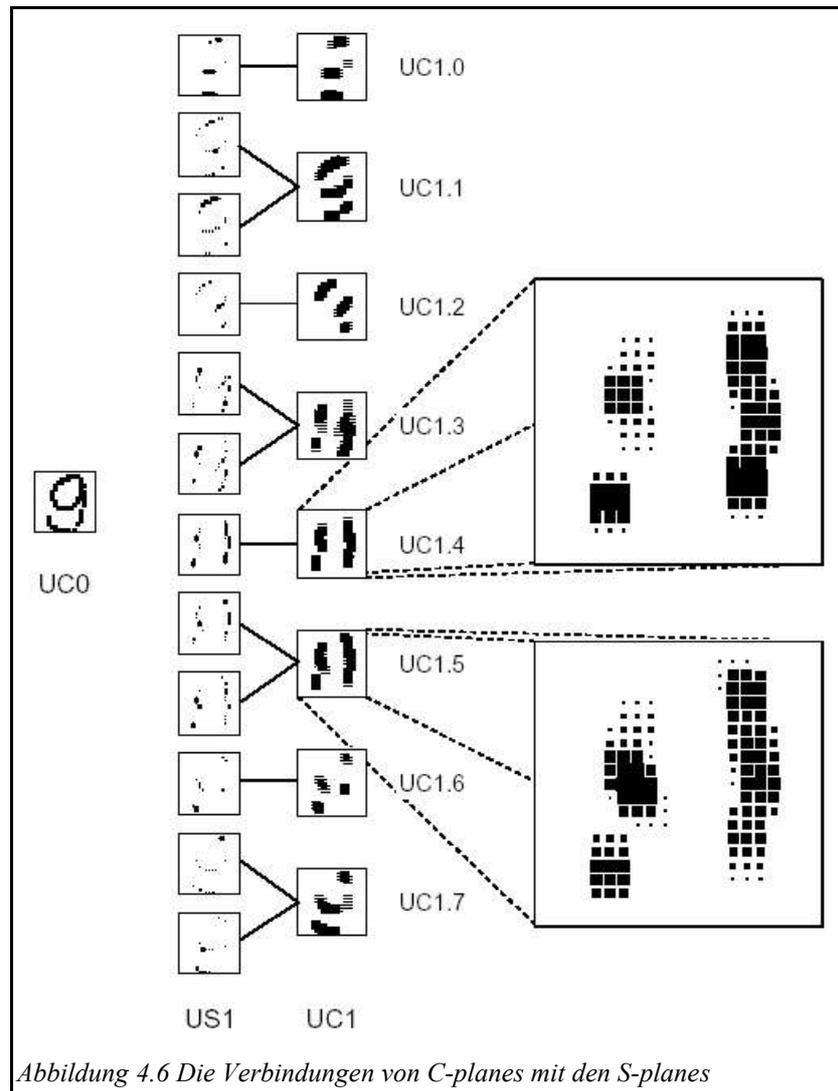
Die Synapsenstärke (Gewicht) jeder S-cell einer S-plane ist gleich groß. Da jede S-cell in dem selben S-plane das gleiche „Merkmal“ erkennen soll, müssen auch die Gewichte mit denen die S-cells an ihre Eingangsfelder gebunden sind gleich groß sein. Dieses wird „weight sharing“ (Aufteilung der Gewichte) genannt. Der Grund dafür ist, daß egal wo ein erlerntes „Merkmal“ im Eingangsfeld auftaucht, es immer gleich gut erkannt werden soll. „Weight sharing“ ist ein Grund weswegen das Neocognitron verschobene und deformierte Muster erkennt, aber nicht der einzige Grund. Die Größe der Gewichte (Synapsenstärke) wird beim Trainieren des Neocognitrons zum Erkennen der Zahlen vergeben.

4.6 Erster Layer Complex Cell

Fukushimas Interpretation der Complex Cell (C-cell) basiert direkt auf dem Modell von Hubel und Wiesel. Die C-cell erhält nur Informationen über Verbindungen zu vorherigen S-cells. Eine Aktivität in einer S-cell würde so eine Reaktion in jeder C-cell auslösen, die mit dieser S-cell verbunden ist (Abbildung 4.5). Die Eingangsfelder der C-cells decken vollständig den vorhergehenden S-plane (oder S-planes, wenn ein C-plane mit mehreren S-planes verbunden ist) ab.

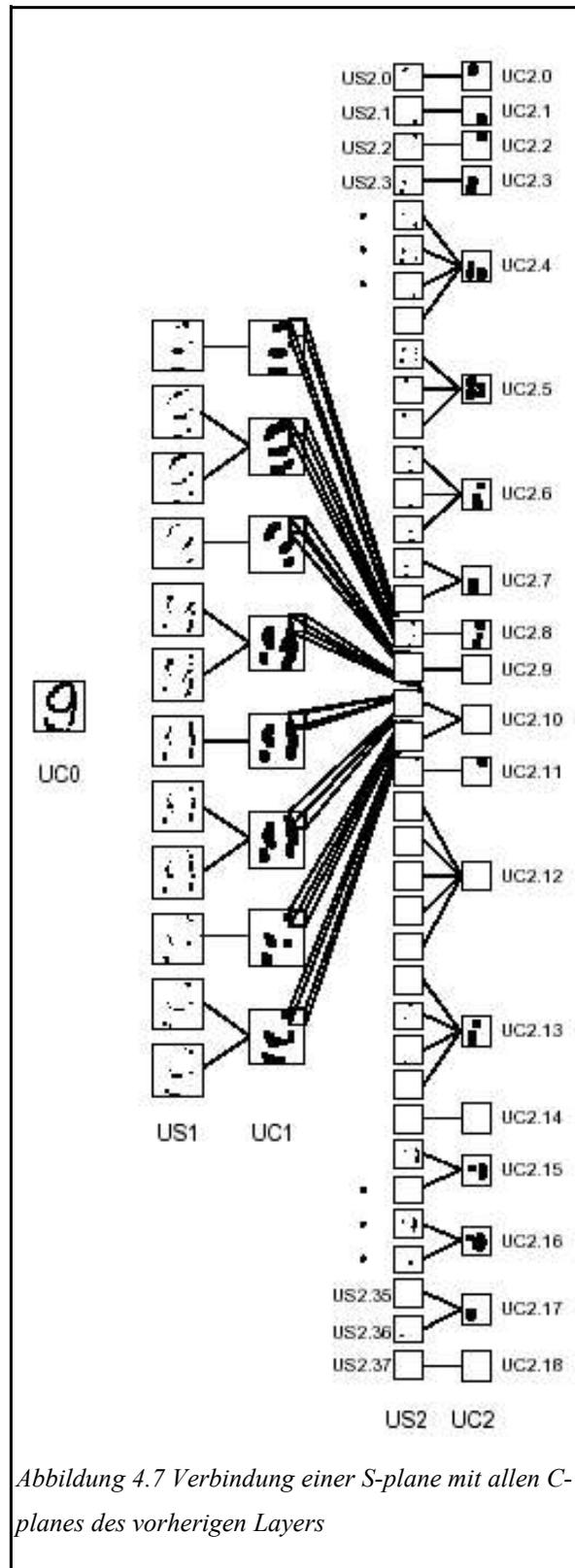


Dabei überlappen die Eingangsfelder der C-cell, wie auch die S-cells in ihren Eingangsfeldern und werten es komplett aus. Kein C-Plane ist mit dem (oder den) selben vorhergehenden S-plane verbunden (Abbildung 4.6). Durch die Verbindung von zwei oder mehreren S-planes mit einem C-plane, kombiniert das C-plane „Merkmale“ die ein einzelnes S-plane nicht erhalten könnte. Auch die Eingangsfelder der C-cells sind mit Gewichten an die C-cells gebunden, allerdings werden diese nicht angelernt, sondern von vornherein fest vorgegeben. Dieses gilt für alle C-planes im gesamten Neocognitron.



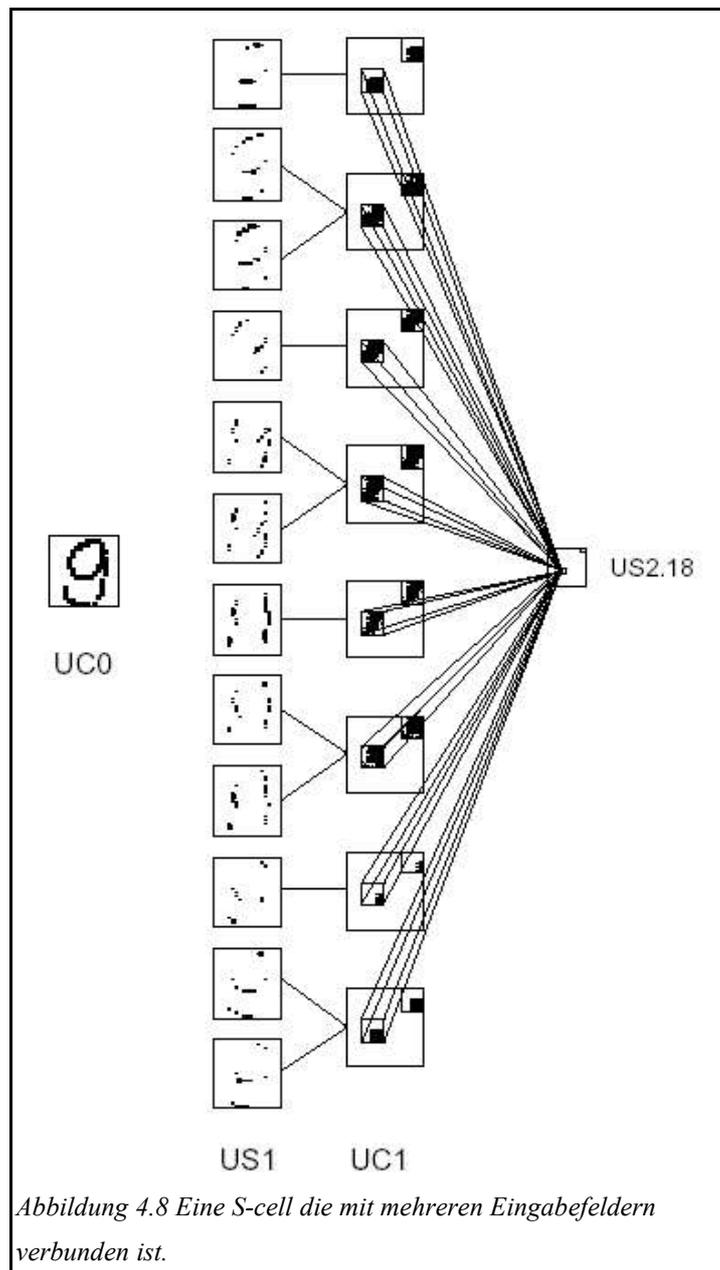
4.7 Layer 2 und folgende

Die S-cells im zweiten Layer US2 des Neocognitrons beziehen ihre Eingangsinformationen von allen UC1 C-planes (Abbildung 4.7). Im vorhergehenden Layer reagierte jedes S-plane nur auf ein bestimmtes 3x3 Pixel großes „Merkmal“. In diesem Layer ist die Funktion der S-cells in US2 das Erkennen von „Merkmalen“ die zusammengesetzt sind aus den einfachen „Merkmalen“ des vorherigen Layers. Diese „zusammengesetzten Merkmale“ mußten vorher für den zweiten Layer angelernt werden, genauso wie die „einfachen Merkmale“ im ersten Layer. Eine genauere Beschreibung des Anlernens erfolgt in einem späteren Abschnitt.



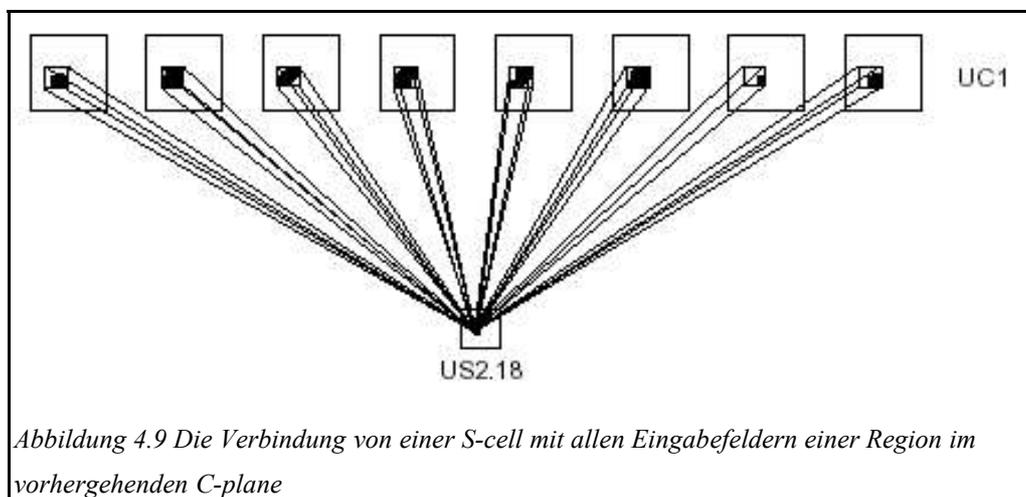
Wie die S-cells im ersten Layer, so sind auch die S-cells in einem S-plane im zweiten Layer alle mit dem gleichen Gewichten mit einem Eingangsfeld verbunden. Da jede S-cell im zweiten Layer aber mit jedem C-plane aus dem vorherigen Layer verbunden ist, hat eine S-cell nicht nur ein

Eingabefeldern sondern mehrere. Damit verfügt eine S-cell auch über eine Menge von Gewichten. Jede dieser Menge von Gewichten verbindet damit eine S-cell mit einer bestimmten Region in allen vorhergehenden C-planes (Abbildung 4.8 und 4.9).

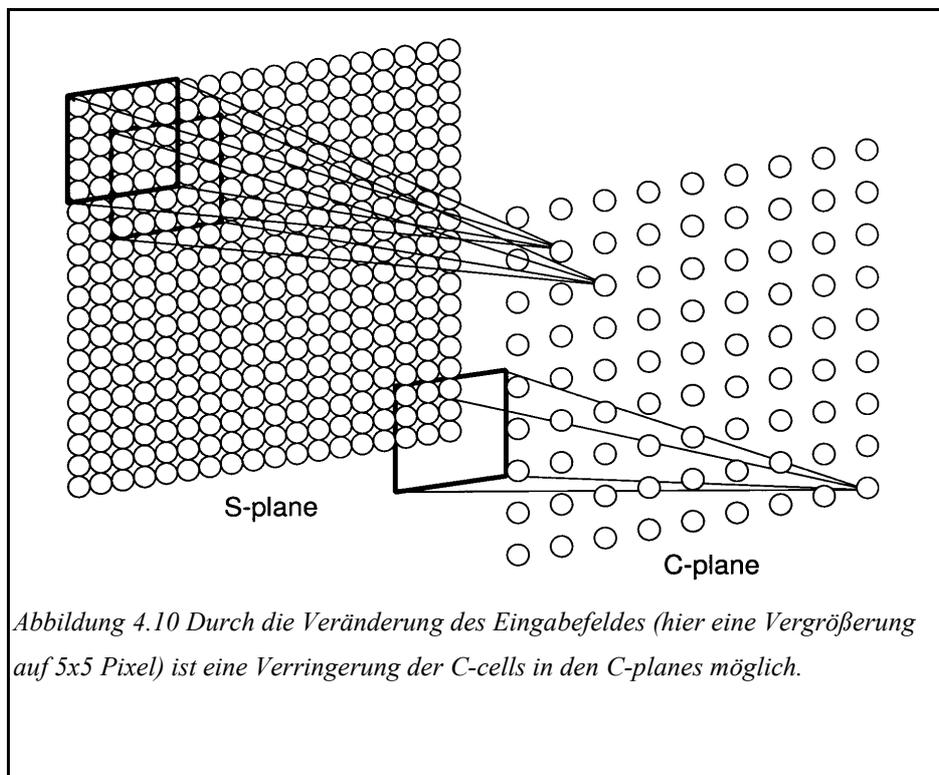


Alle S-cells einer S-plane decken damit alle C-planes in einem Layer ab und untersuchen diese nach einem bestimmten erlernten „zusammengesetzten Merkmal“. In höheren Layern nimmt so sukzessive die Komplexität der „Merkmale“ zu. Im letzten Layer ist es dann die Aufgabe der S-cells „zusammengesetzte Merkmale“ zu erkennen die den gelernten Zahlen 0 bis 9 gleichen.

Im echten Leben sehen handgeschriebene Zahlen von der selben Person fast niemals gleich aus, es gibt einen gewissen Spielraum in der Schreibweise. Zur Erkennung dieser „deformierten“ Zahlen dienen die C-cells. In Abbildung 4.6 kann deutlich erkannt werden, daß in den C-planes die „Merkmale“ aus den S-planes unscharf abgebildet werden. Das kann dadurch erklärt werden, daß durch die Überlappung der Eingangsfelder in einem C-plane die Reaktion einer S-cell von mehreren nebeneinander liegenden C-cells ausgewertet wird. Durch diese unscharfe Abbildung der „Merkmale“ in den C-planes können die nachfolgenden S-cells auch „Merkmale“ erkennen, die nicht ganz das Aussehen haben. Denn durch die Unschärfe wird ein „Merkmal“ quasi mehrmals nebeneinander abgebildet. Damit wird die Chance größer, daß in den richtigen Positionen für die Erkennung genug aktive C-cells vorhanden sind, damit die S-cell reagiert. Die Abweichung (Deformation) des „Merkmals“ darf allerdings auch nicht zu groß sein. Durch die Anbindung einer S-cell an mehrere Eingangsfelder (Abbildung 4.8 und 4.9) zur Auswertung der vorhergehenden C-planes ist nicht nur die Erkennung eines „Merkmals“ möglich, sondern auch der Abstand der „Merkmale“ zu einander. Mit der Unschärfe in den C-planes ist so auch eine Erkennung von Mustern möglich, die in gewissen Grenzen in der Größe variieren.



Eine zusätzliche Aufgabe der C-planes ist die Verringerung der Größe der C-planes und S-planes. Während die S-planes immer die Größe der vorhergehenden C-planes haben, verringert sich die Größe der C-planes ab dem zweiten Layer. Dieses wird durch eine Anpassung des Eingabefeldes erreicht, so daß mehr S-cells auf einmal ausgewertet werden können und die Anzahl der C-cells im C-plane sinkt (Abbildung 4.10). Im letzten Layer besteht ein C-plane dann nur noch aus einer C-cell, die dann nur mit einer der angelernten Zahlen korrespondiert (Abbildung 4.11).



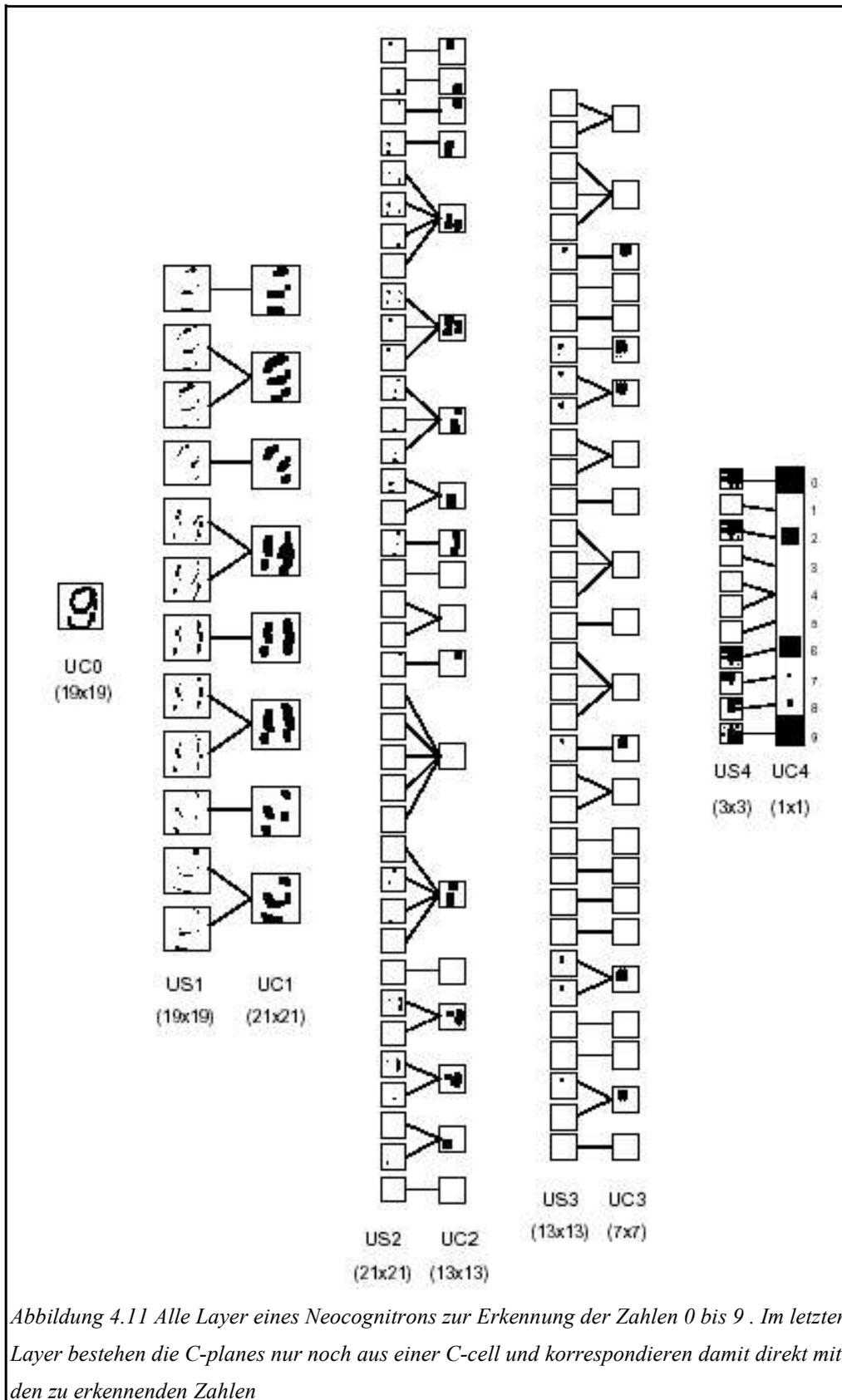


Abbildung 4.11 Alle Layer eines Neocognitrons zur Erkennung der Zahlen 0 bis 9 . Im letzten Layer bestehen die C-planes nur noch aus einer C-cell und korrespondieren damit direkt mit den zu erkennenden Zahlen

4.8 Training des Neocognitrons

Es gibt zwei Möglichkeiten das Neocognitron zu trainieren (Muster anzulernen). Das Lernen mit Lehrer (supervised learning) und das Lernen ohne Lehrer (unsupervised learning). Bei beiden Methoden des Anlernens ist es nötig, alle Layer nacheinander anzulernen, da die S- oder C-cells des nachfolgenden Layers nur das Lernen kann, was im vorhergehenden Layer erkannt wurde. Auch die Erhöhung der Gewichte mit denen die S-cells an ihre Eingangsfelder gebunden sind verläuft bei beiden Methoden nach dem gleichen Prinzip und ähnelt der Hebb'schen Lernregel. Nur die S-cells verfügen allerdings über Gewichte die verstärkt werden können, die Gewichte der C-cells werden bei der Programmierung festgelegt und sind im laufenden Programm nicht änderbar.

4.8.1 Lernen ohne Lehrer

Beim Lernen ohne Lehrer werden Muster angelernt, ohne zusätzliche Hilfe für das Neocognitron. Dem Neocognitron wird ein Muster vorgelegt. Eine Zerlegung dieses Musters in seine „Merkmale“ erfolgt durch einen einfachen Mechanismus. Es werden alle S-cells des ersten Layers auf ihre Reaktion ausgewertet. Dazu wird die Menge der Gewichte aller S-cells zu ihren Eingangsfeldern mit kleinen Zufallszahlen vorbelegt, wobei alle S-cells die gleichen Gewichte erhalten. Die S-cells mit der stärksten Reaktion, deren Eingangsfelder nicht an der gleichen Position sind, werden ausgewählt und „Seed cells“ genannt. Die Anzahl der „Seed cells“ entspricht der Anzahl der S-planes im Layer, denn jeder S-plane ist nur für die Erkennung eines Merkmals zuständig. Beim folgenden Anlernen werden nur die Gewichte dieser „Seed cells“ verstärkt. Nach dem Anlernen erhalten alle S-cells desselben S-planes die gleichen Gewichte für ihr Eingangsfeld, damit alle S-cells eines S-planes nur auf das angelernte „Merkmal“ reagieren (weight sharing). Dieses stellt sicher, daß unterschiedliche S-planes auch nur auf unterschiedliche „Merkmale“ reagieren.

Bei dieser Methode des Anlernens gibt es allerdings ein Problem: Es ist nicht kontrollierbar, welche „Merkmale“ angelernt werden und ob es wirklich ein markantes „Merkmal“ des Musters ist. Im ungünstigsten Fall lernt das Neocognitron „Merkmale“, die sich alle nur unwesentlich unterscheiden oder sogar das gleiche „Merkmal“ nur an unterschiedlichen Positionen. Ein so trainiertes Neocognitron wäre zur Erkennung untauglich. Denn das Neocognitron verfügt über die Möglichkeit, auch deformierte Muster zu erkennen und könnte daher Muster, die alle über solche ähnlichen „Merkmale“ verfügen, nicht mehr unterscheiden. Die Fähigkeit zur Unterscheidung von

Mustern wird auf ein einziges „Merkmal“ reduziert. Dadurch sind völlig unterschiedliche Muster, die zufällig dieses „Merkmal“ besitzen, nicht mehr unterscheidbar. Fukushima selbst stellte fest, daß das Lernen ohne Lehrer (also ein Selbst-organisierendes Neocognitron) keine guten Ergebnisse brachte. Er entwickelte deswegen eine verbesserte Methode des Anlernens: „Lernen mit Lehrer“.

4.8.2 Lernen mit Lehrer

Beim Lernen mit Lehrer wird die Auswahl der „Merkmale“ eines Musters nicht von dem Neocognitron selbst vorgenommen, sondern erfolgt durch einen „Lehrer“ (Menschen). Die Abbildungen 4.12, 4.13, 4.14 und 4.15 zeigen die „Merkmale“, die Fukushima für die vier Layer seiner Handschriftenerkennung der Zahlen 0 bis 9 auswählte. Dabei werden dem Neocognitron aufeinander folgend die verschiedenen Merkmale „präsentiert“ und so die Layer sukzessive angelernt. Das eigentliche Anlernen ist dabei mit dem Lernen ohne Lehrer identisch. Der Unterschied zum Lernen ohne Lehrer besteht darin, daß die „Seed-cell“ von dem Lehrer festgelegt wird durch die Festlegung des Eingabefeldes, in dem das Merkmal „präsentiert“ wird. Der Vorteil dieser Form des Anlernens ist nicht nur die Auswahl der „Merkmale“, sondern auch die Kontrolle des Lernfortschritts nach jedem Layer.

Der Nachteil dieser Art des Anlernens ist aber der gewaltige Aufwand zur Vorbereitung der „Merkmale“ für jeden Layer. Dieser Aufwand wird an Hand der Abbildungen 4.12 – 4.15 verdeutlicht.

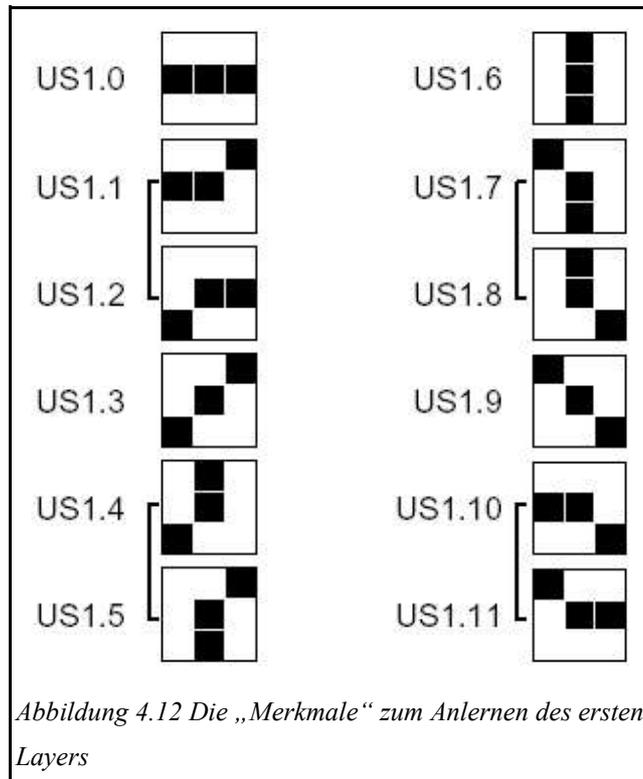
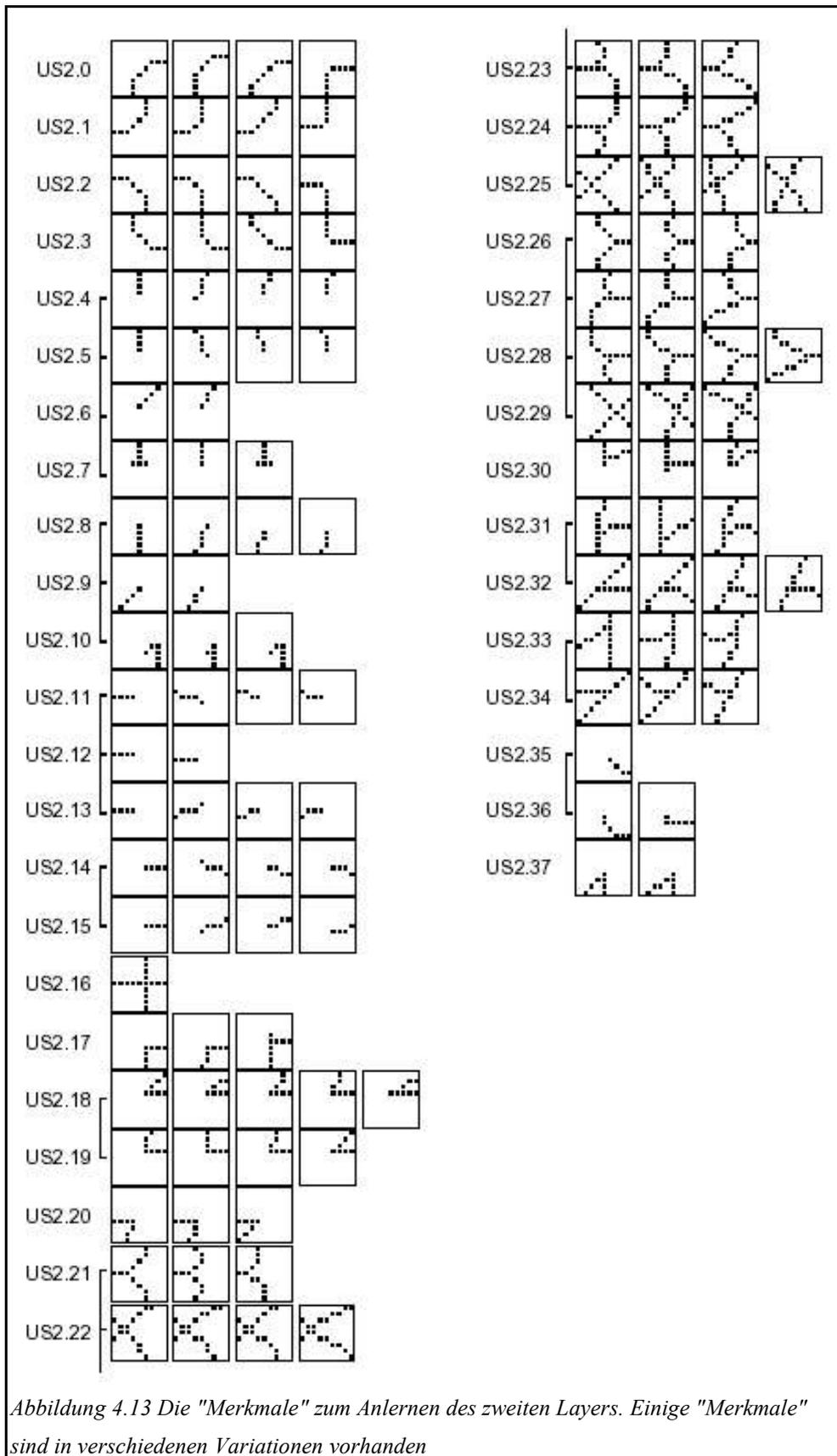


Abbildung 4.12 Die „Merkmale“ zum Anlernen des ersten Layers



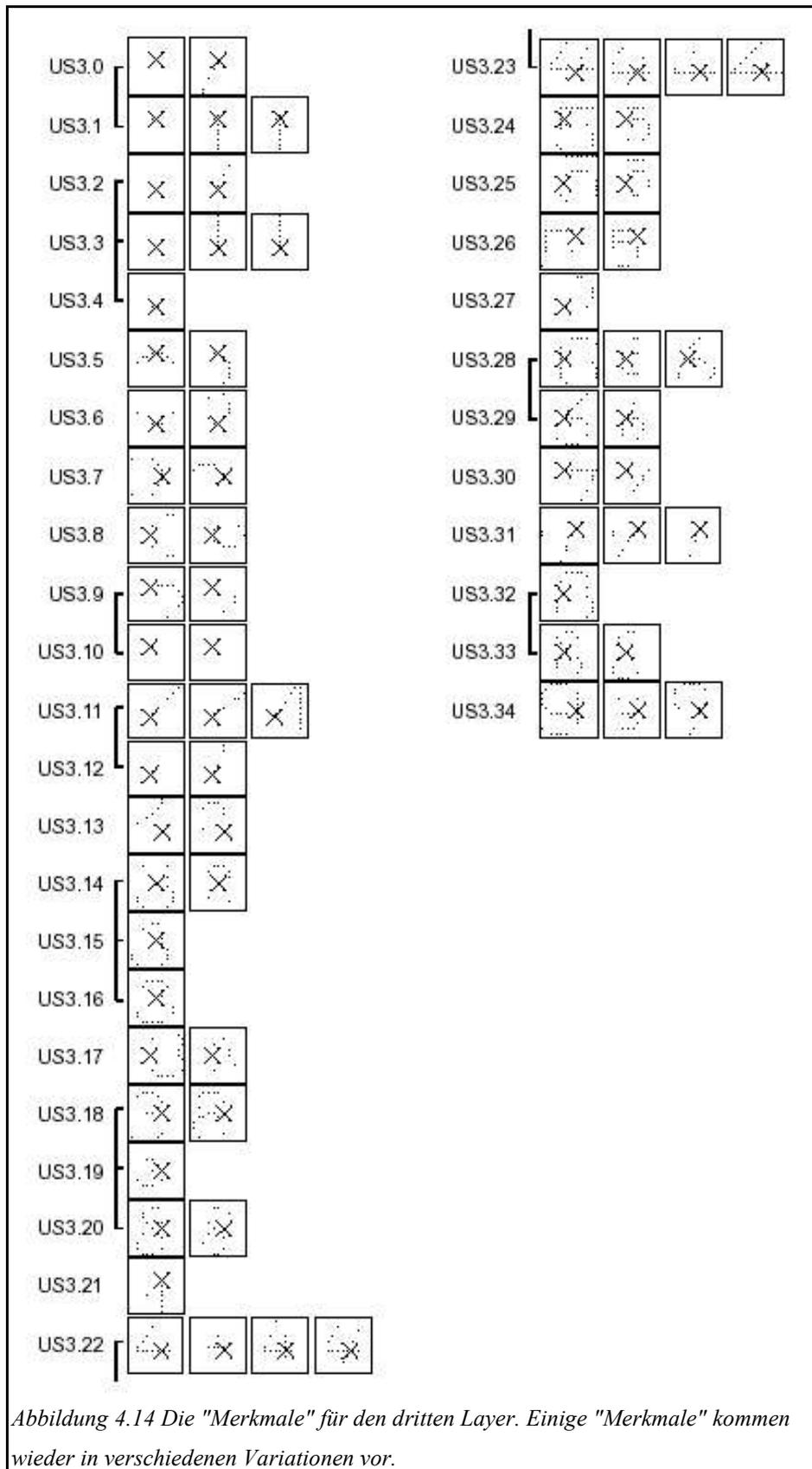
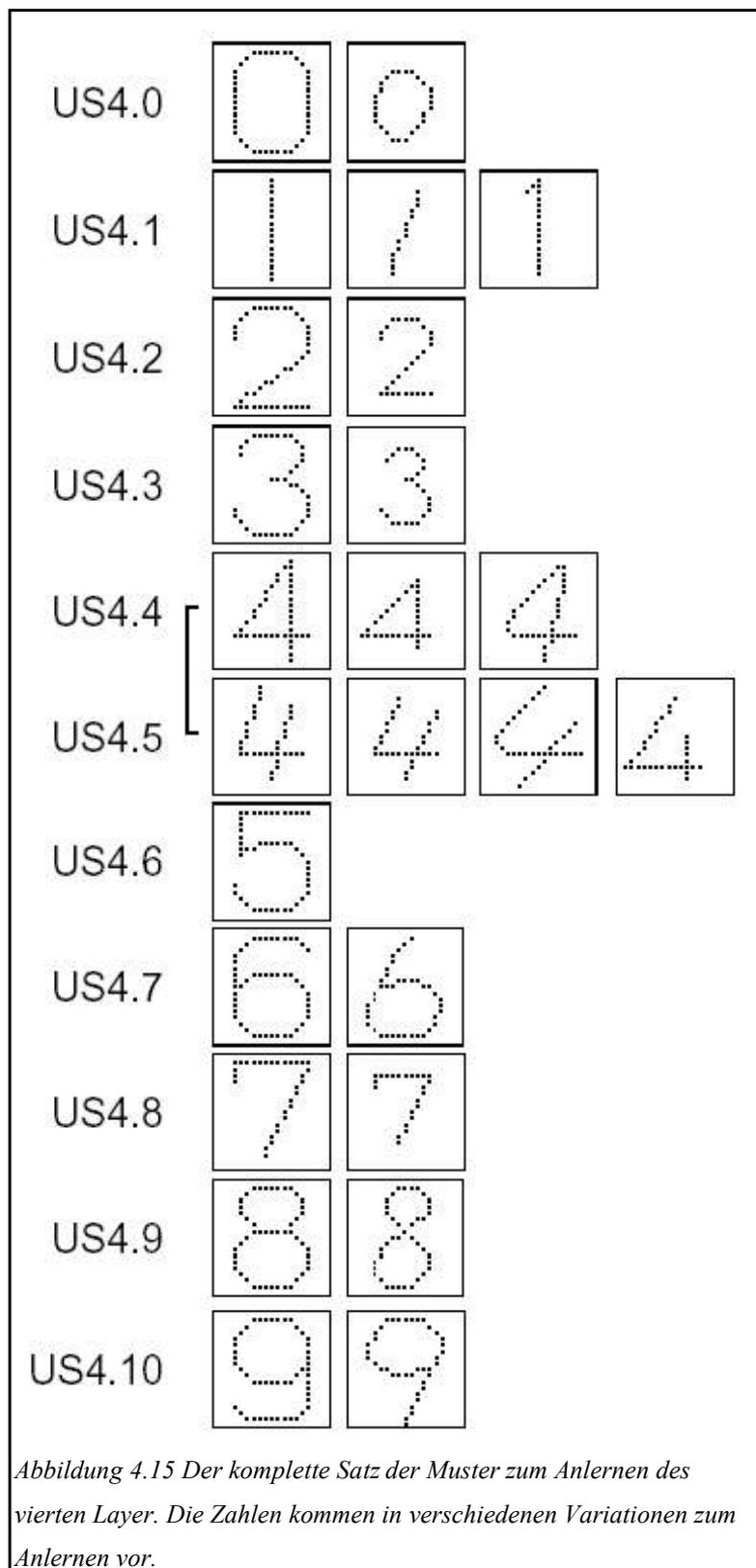


Abbildung 4.14 Die "Merkmale" für den dritten Layer. Einige "Merkmale" kommen wieder in verschiedenen Variationen vor.



4.9 Die Morphologie des Neocognitrons

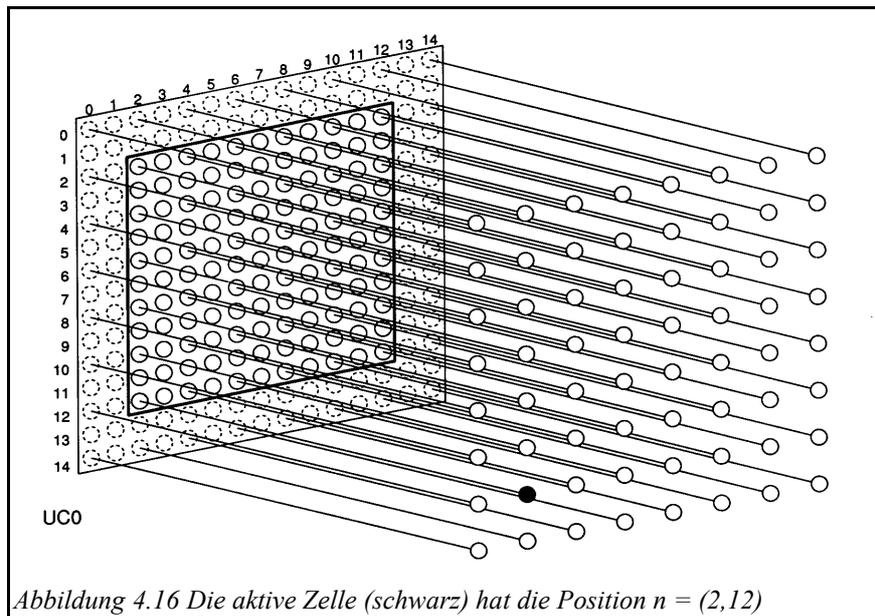
Es gibt drei unterschiedliche Elemente zur Verarbeitung von Informationen im Neocognitron. Die S-cell, C-cell und die V-cell. Jede S- oder C-cell wird durch vier Informationen spezifiziert :

- Type S oder C für S- oder C-cell.
- Layer l , zu dem die S- oder C-cell gehört.
- Plane (S-plane oder C-plane) k , zu dem die S- oder C-cell gehört.
- Position n , die eine S- oder C-cell in einem Plane einnimmt.

Die Werte (der Wert einer Reaktion bei Anregung) einer S-cell oder C-cell hat damit folgende Notation: $u_{S_l}(n,k)$ oder $u_{C_l}(n,k)$. V-cells liefern Informationen an die S-cells über die Stärke der Aktivität in deren Eingangsfeldern. Es wird nur eine V-plane pro Layer benötigt um alle Eingangsfelder der S-cells auszuwerten. Eine V-cell ist folgendermaßen spezifiziert :

- Typ (hier V für V-cell).
- Layer l , zu dem die V-plane gehört.
- Position n , die die V-cell in einem Plane einnimmt.

V-cells benutzen folgende Notation: $u_{V_l}(n)$. Die möglichen Werte der Parameter l , k und n werden durch den Aufbau eines Neocognitron festgelegt. Der Parameter n ist ein zweidimensionaler Vektor. Dieser Vektor beschreibt die Position im Eingangsfeld (Abbildung 4.16).



4.10 Die synaptische Organisation des Neocognitrons

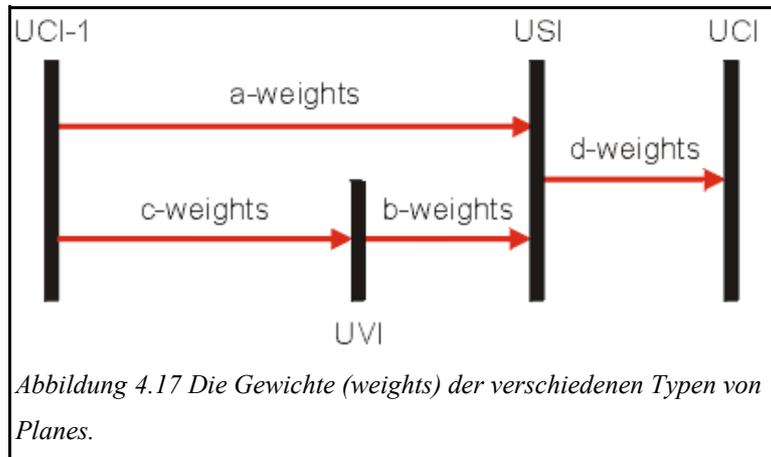
Das Neocognitron ist in Schichten (Layer) von sich abwechselnden S- und C-planes aufgebaut. Nur benachbarte Planes sind direkt miteinander verbunden. Eine S-cell ist über ihr Eingangsfeld mit allen C-planes des vorherigen Layers verbunden. Eine C-cell ist hingegen nur mit bestimmten S-planes verbunden. Diese Verbindung wird Gewicht genannt und ist folgendermaßen spezifiziert :

- Der Layer l , der S-plane mit dem die C-cell verbunden ist.
- Die Nummer k , des S-planes.
- Die Nummer K , von dem C-plane, zu dem die C-cell gehört.
- Die Position v , des Eingangsfeldes in der C-plane.

Ein Gewicht (weight) von einer C-cell zu einem S-plane hat damit folgende Notation:

$a_l(v, K, k)$. Jede S-cell erhält auch Informationen von einer V-cell. Die Stärke der Gewichte

(weights) wird in $b_l(k)$ festgelegt. Die Abbildung 4.17 zeigt diese Gewichte (weights) :



Die Verbindung zwischen C-planes und jeder V-cell sind festgelegt als $c_i(v)$. Wobei v die Position des Eingabefeldes in einem C-plane ist.

Die Verbindung von S-cells zu eine C-plane hat folgende Notation: $d_i(v)$. „ v “ ist dabei die Position des Eingabefeldes im S-plane. Da ein C-plane mit mehreren S-planes verbunden sein kann, gibt es einen Faktor $j(K,k)$, der die Verbindung beschreibt. Wenn der S-plane k mit dem C-plane K verbunden ist, dann ist $j(K,k) > 0$ ansonsten ist $j(K,k) = 0$.

Nur die Gewichte $a_i(v,K,k)$ und $b_i(k)$ werden beim Training des Neocognitrons verstärkt.

Die Gewichte $c_i(v)$ und $d_i(v)$ sind fest, d.h. die Gewichte werden vor dem Anlernen festgelegt.

4.11 Die mathematische Beschreibung des Neocognitrons

Es gibt einige Neuronale Netze die sich sehr einfach mathematisch beschreiben lassen. Das Neocognitron gehört leider nicht zu diesen.

Dieser Abschnitt beschreibt die Reaktionen der S-, C-, und V-cell mathematisch.

Der Wert bei einer Reaktion einer S-cell im S-plane k im Layer l berechnet sich nach folgendem Ausdruck:

$$u_{sl}(n, k) = r_l \cdot \varphi \left[\frac{1 + \sum_{L=1}^{L_{cl-1}} \sum_{v \in A_l} a_l(v, K, k) \cdot u_{cl-1}(n+v, K)}{1 + \frac{r_l}{r_l+1} \cdot b_l(k) \cdot u_{vl}(n)} - 1 \right] \quad (12)$$

Alle Terme des Ausdrucks sind im vorangegangenen Abschnitt erklärt worden, außer :

L_{cl-1} = Nummer des C-plane im vorherigen Layer,

A_l = Eingangsfeld im C-plane

r_l = Selektivitätsparameter (Toleranz) und

$$\varphi(x) = \varphi = \begin{cases} 0 & \text{wenn } x < 0 \\ x & \text{wenn } x \geq 0 \end{cases} \quad (13) \text{ als Grenzfunktion .}$$

Der Selektivitätsparameter r_l bestimmt, wie nahe das zu erkennende „Merkmal“ mit dem angelernten „Merkmal“ übereinstimmen muß um noch erkannt zu werden.

Der Wert einer C-cell kann mit folgendem Ausdruck berechnet werden :

$$u_{cl}(n, k) = \psi \left[\sum_{L=1}^{L_{sl}} j(k, K) \cdot \sum_{v \in D_l} d_l(v) \cdot u_{sl}(n+v, K) \right] \quad (14)$$

mit

L_{sl} = Nummer des S-plane im vorherigen Layer,

D_l = Eingangsfeld im S-plane und

$$\psi(x) = \Psi(x) = \frac{\varphi(x)}{1 + \varphi(x)} \quad (15) \text{ als Sättigungsfunktion.}$$

Die Sättigungsfunktion limitiert den Ausgangswert einer C-cell auf den Bereich [0;1].

V-cells haben einen hemmenden Effekt bei dem Ausgangswert einer S-cell. Der Wert einer V-cell berechnet sich mit folgendem Ausdruck :

$$u_{vl}(n) = \sqrt{\sum_{L=1}^{K_{Cl}-1} \sum_{v \in A_l} c_l(v) \cdot u_{cl-1}^2(n+v, K)} \quad (16)$$

Die Gewichte $a_l(v, K, k)$ und $b_l(k)$ werden beim Anlernen des Neocognitrons folgendermaßen erhöht:

$$\begin{aligned} \Delta a_l(v, K, \hat{k}) &= q_l \cdot c_l(v) \cdot u_{cl-1}(\hat{n} + v, K) \\ \Delta b_l(\hat{k}) &= q_l \cdot u_{vl}(\hat{n}) \end{aligned} \quad (17)$$

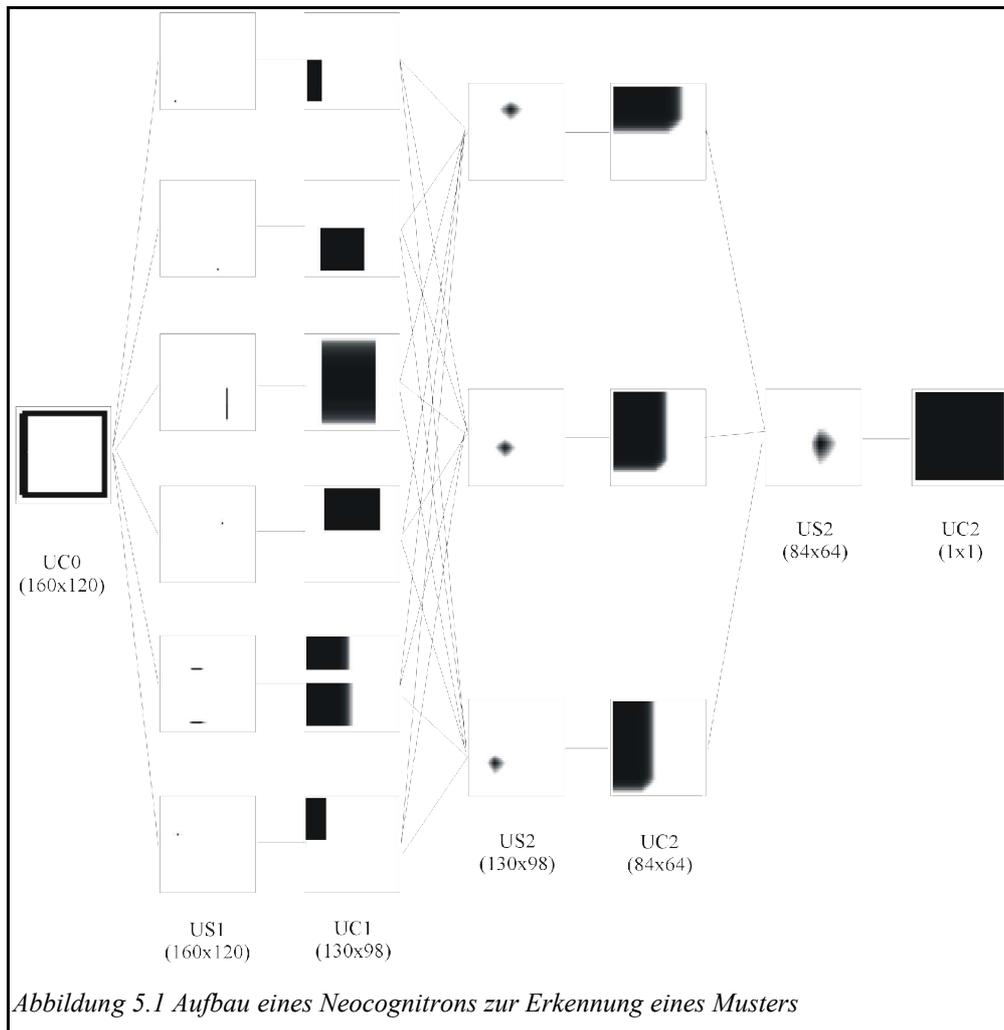
mit \hat{k} = Nummer des Planes der Seed Cell
 \hat{n} = Seed Cell Position
 und q_l = Lern-Koeffizient

5 Anwendung des Neocognitron in dieser Arbeit

5.1 Der Aufbau

Das Neocognitron von Fukushima zur Handschriftenerkennung wurde nur für den Zweck entwickelt, die Zahlen 0 bis 9 zu erkennen und zu unterscheiden. Diese Spezialisierung wird bei der Zusammenfassung von einfachen „Merkmale“ zu komplexen „Merkmalen“ deutlich (Abbildung 4.11): Nur wenn die Eingangsmuster und auch deren „Merkmale“ bekannt sind, kann eine Zusammenfassung zu „komplexen Merkmalen“ stattfinden. Bei einem unbekanntem Muster, kann eine wahllose Zusammenfassung von Merkmalen die Leistungsfähigkeit eines Neocognitrons schwächen.

Das Neocognitron, das im Rahmen dieser Diplomarbeit programmiert wurde, unterscheidet sich deswegen von dem Neocognitron zur Handschriftenerkennung. Es ist nicht auf die Erkennung eines speziellen Musters, sondern auf die Erkennung eines einzelnen allgemeinen Musters ausgelegt. Der hierzu notwendige Aufbau wird in Abbildung 5.1 verdeutlicht.



Durch die Erkennung nur eines Musters benötigt dieses Neocognitron weniger S- und C-planes pro Layer, dafür ist die Anzahl der S- und C-cells in den Planes aber erheblich größer als in dem Neocognitron zur Handschriftenerkennung. Dieses bietet die Möglichkeit, wesentlich komplexere „Merkmale“ gleich zu Beginn anzulernen, ohne sie durch Zusammenfassung von S-planes in C-planes zu erzeugen.

5.2 Die Trainingsmethoden

Im Gegensatz zu dem Neocognitron von Fukushima verfügt das Neocognitron, das im Rahmen dieser Diplomarbeit programmiert wurde, über drei Trainingsmethoden: „Lernen ohne Lehrer“, „Lernen mit Lehrer“ und eine Zwischenform, bei der der erste Layer nach der Methode „Lernen mit Lehrer“ trainiert wird und die anderen Layer mit „Lernen ohne Lehrer“ trainiert werden. Dieses hat den Vorteil, daß der Lehrer (Mensch) nur für den ersten Layer die „einfachen Merkmale“ aus dem Muster extrahieren muß, aber nicht die „komplexeren Merkmale“ für die höheren Layer wie bei der Methode von Fukushima. Dieses vereinfacht das Anlernen von Mustern erheblich. Zusätzlich wurde beim „Lernen ohne Lehrer“ eine kleine Veränderung vorgenommen um zu verhindern, daß gleiche Muster angelernt werden: Die S-cells des ersten Layer verfügen bei der Auswahl der „Merkmale“ alle über dieselben Gewichte zu ihrem Eingangsfeld. Identische „Merkmale“ haben deswegen auch denselben Wert bei der Reaktion der S-cells. Damit können, zur Erhöhung der Effektivität, gleiche Merkmale ausgeschlossen werden.

5.3 Die Parameter des Neocognitrons

Für das Anlernen, aber auch für das Erkennen von Mustern, gibt es einige Parameter die verändert werden können, um bessere Ergebnisse zu erzielen. Die richtigen Parameter auszuwählen ist aber auch eines der größten Probleme beim Neocognitron.

Beim Training (Anlernen) sind folgende Parameter im Neocognitron, das dieser Arbeit zugrunde liegt, unabhängig einstellbar von der Trainingsmethode :

- Anzahl der Lerndurchgänge
- Der Lernfaktor q_l
- Der Selektivitätsfaktor (Toleranz) r_l für jeden Layer

Die Anzahl der Lerndurchgänge gibt die Häufigkeit der Erhöhung der Gewichte an, wobei der Lernfaktor q_l den Betrag (siehe Ausdruck (17)) angibt, um den die Gewichte erhöht werden. Diese beiden Parameter sind nur für das Anlernen von Bedeutung. Der Selektivitätsfaktor (die Toleranz) r_l ist sowohl beim Anlernen als auch beim Erkennen von Mustern von Bedeutung. Denn beim Anlernen kann jeder Layer nur das Anlernen, was der vorherige Layer erkannt hat. Der Faktor r_l

bestimmt, wie stark die Reaktion der S-cells beim Erkennen in jedem Layer ist. Ein hoher Selektivitätsfaktor hemmt die Reaktion der S-cells, damit reagiert die S-cell nur noch wenn, das „Merkmal“ genau mit dem Angelernten übereinstimmt. Bei einem niedrigen Selektivitätsfaktor werden auch „Merkmale“ erkannt, die sich von dem angelernten „Merkmal“ unterscheiden. Wenn „Merkmale“ mit einem niedrigen Selektivitätsfaktor erkannt werden, so werden auch die S-cells ausgewertet, die bei einem höheren Selektivitätsfaktor nicht berücksichtigt werden, d.h. daß gleiche „Merkmale“ je nach Höhe des Selektivitätsfaktor anders erkannt werden.

5.4 Das Anlernen und Erkennen von dreidimensionalen Mustern

5.4.1 Aufbau der Ergebnistabellen

Die nachfolgenden Tabellen zeigen die Ergebnisse bei der Erkennung der folgenden dreidimensionalen Muster:

- A = Würfel
- B = Quader
- C = Kugel
- D = Pyramide

Die Tabellen haben dabei folgenden Aufbau :

Am Anfang der Tabelle stehen die Einstellungen für das Anlernen der Muster (Lerndurchgänge, Lernfaktor). Die anderen beiden Parameter (init. der a-weights und Kontrast) haben keine weitere Bedeutung für das Anlernen oder Erkennen. Die Toleranzwerte in der Tabelle wurden sowohl beim Anlernen als auch beim Erkennen benutzt. Die Zahlen in der Spalte „Erkennung“ gehen aus der nicht linearen Sättigungsfunktion (siehe Ausdruck (15)) hervor. Sie können deswegen nicht als Prozentzahl interpretiert werden. Sie geben nur ungefähr an, wie gut ein Muster erkannt wurde. Die Ergebnisse stellen nur einen repräsentativen Ausschnitt der gesamten Ergebnisse dar. Alle Ergebnisse würden den Rahmen dieser Diplomarbeit sprengen.

5.4.2 Ergebnisse: Toleranz

Einstellungen								
Lernfaktor q			1					
Lerndurchgänge			100000					
Init. Der a-weights			Null					
Kontrast			200					
Objekte			A – Würfel					
			B – Quader					
			C – Kugel					
			D – Pyramide					

Toleranz			Lernobjekt	Erkennung			
Layer 1	Layer 2	Layer 3		A	B	C	D
1	1	1	A	0.9995	0.9997	0.9997	0.9996
2	2	2	A	0.9985	0.9985	0.9789	-
3	3	3	A	0.9980	0.9858	-	-
4	4	4	A	0.9903	0.9166	-	-
5	5	5	A	0.9954	-	-	-
6	6	6	A	0.9784	-	-	-
7	7	7	A	0.9886	-	-	-
8	8	8	A	0.9638	-	-	-
9	9	9	A	0.7121	-	-	-
1	2	3	A	0.9995	0.9993	0.9977	0.9913
1	3	2	A	0.9993	0.9995	0.9987	0.9987
2	1	3	A	0.9994	0.9994	0.9981	-
2	3	1	A	0.9993	0.9984	0.9331	-
3	1	2	A	0.9993	0.9995	0.9983	0.9982
3	2	1	A	0.9993	0.9992	0.9976	-
2	3	4	A	0.9967	0.9883	-	-
2	4	3	A	0.9975	0.9955	-	-
3	2	4	A	0.9986	0.9980	-	-
3	4	2	A	0.9968	0.9935	-	-
4	2	3	A	0.9982	0.9972	-	-
4	3	2	A	0.9956	0.9938	-	-
3	4	5	A	0.9921	0.7986	-	-
3	5	4	A	0.9949	0.9176	-	-
4	3	5	A	0.9963	0.9765	-	-
4	5	3	A	0.9967	-	-	-
5	3	4	A	0.9945	0.9891	-	-
5	4	3	A	0.9976	-	-	-
4	5	6	A	0.9874	0.8877	-	-
4	6	5	A	0.9945	-	-	-
5	4	6	A	0.9892	-	-	-
5	6	4	A	0.9930	-	-	-
6	4	5	A	0.9921	0.7866	-	-
6	5	4	A	0.9963	-	-	-
5	6	7	A	0.9924	-	-	-

Tabelle 5.1 Auswertung von „Lernen mit Lehrer nur im ersten Layer“

In der Tabelle 5.1 wird durch verschiedene Selektivitätsfaktoren (Toleranzwerte) r_l für alle drei Layer versucht, die beste Einstellung herauszufinden. Es zeigt sich, daß mit sehr niedrigen Toleranzwerten das Neocognitron nicht mehr in der Lage ist, zwischen verschiedenen Mustern zu unterscheiden. Auch bei sehr hohen Toleranzwerten sinkt die Erkennungsleistung des angelegten Musters. Das beste Ergebnis erzielt die Einstellung „5, 4, 3“. In den Tabellen 5.2, 5.3 und 5.3 wird gezeigt, daß auch bei den verschiedenen Trainingsmethoden mit dieser Einstellung gute Ergebnisse erzielt werden.

Einstellungen				
Toleranz Layer 1				5
Toleranz Layer 2				6
Toleranz Layer 3				7
Lernfaktor q				10
Anzahl der Lerndurchgänge				10
Init. Der a-weights				Null
Kontrast				200
Objekte	A – Würfel			
	B – Quader			
	C – Kugel			
	D – Pyramide			

Lernobjekt	Erkennung			
	A	B	C	D
A	0.9866	-	-	-
B	-	0.9867	-	-
C	-	-	0.9796	-
D	-	-	-	0.9667

Tabelle 5.2 "Lernen mit Lehrer im ersten Layer" bei der Erkennung der vier Muster

Einstellungen				
Toleranz Layer 1				5
Toleranz Layer 2				6
Toleranz Layer 3				7
Lernfaktor q				10
Anzahl der Lerndurchgänge				10
Init. Der a-weights				Null
Kontrast				200
Objekte				A – Würfel B – Quader C – Kugel D – Pyramide

Lernobjekt	Erkennung			
	A	B	C	D
A	0.9874	-	-	-
B	-	0.9794	-	-
C	-	-	0.9924	-
D	-	-	-	0.9887

Tabelle 5.3 Auswertung "Lernen mit Lehrer"

Einstellungen				
Toleranz Layer 1				5
Toleranz Layer 2				6
Toleranz Layer 3				7
Lernfaktor q				10
Anzahl der Lerndurchgänge				10
Init. Der a-weights				Null
Kontrast				200
Objekte				A – Würfel B – Quader C – Kugel D – Pyramide

Lernobjekt	Erkennung			
	A	B	C	D
A	0.9893	-	-	-
B	0.9933	0.9834	-	-
C	-	-	0.9757	-
D	-	-	-	0.9797

Tabelle 5.4 Auswertung "Lernen ohne Lehrer"

In Tabelle 5.4 zeigt sich deutlich das Problem bei „Lernen ohne Lehrer“: Bei einem angelernten Quader wird auch ein Würfel erkannt, sogar mit einem höheren Erkennungswert. Dieses liegt an der schlechten Wahl der „Merkmale“ durch das Neocognitron.

5.4.3 Ergebnisse: Lerndurchgänge und Lernfaktor

Eines der größten Probleme des Neocognitrons ist die Zeit die es zum Anlernen von Mustern benötigt. Normalerweise hat der Lernfaktor q einen Wert im Bereich $[0,1 ; 1]$ damit den Gewichten die Möglichkeit gegeben wird, sehr feine Abstufungen zu erreichen. Die Gewichte müssen aber eine gewisse Mindestgröße aufweisen. Diese beim Neocognitron herauszufinden ist nur durch ausprobieren möglich. In der Literatur wird vorgeschlagen, den Lernfaktor q auf 0,1 zu setzen und 100.000 Lerndurchgänge vorzunehmen. Dieses kann je nach Computer und Größe des Neocognitrons sehr lange dauern. Fukushima versuchte diesem Problem durch „Lernen mit Lehrer“ zu begegnen. In diesem Fall muß nicht auf das Anlernen des gesamten Neocognitrons gewartet werden, sondern es ist eine Aussage über die Erkennungsleistung nach jedem Layer möglich.

In Tabelle 5.5 wird versucht gute Ergebnisse beim Anlernen durch Parameter zu erreichen die außerhalb von denen liegen, die in der Literatur vorgeschlagen werden. Bei einem Lernfaktor $q=10$ und ebenfalls zehn Lerndurchgängen werden gute Ergebnisse erzielt. Einige Parameter zeigen noch bessere Ergebnisse, allerdings haben hier nicht aufgeführte Tests gezeigt, daß der Lernfaktor $q=10$ und zehn Lerndurchgängen der beste Kompromiss zwischen der Zeit und der Qualität des Anlernens ist.

5.5 Das Anlernen und Erkennen von zweidimensionalen Mustern

Dreidimensionale Muster sind für das Neocognitron, das dieser Arbeit zugrunde liegt, nur eine Art deformierte Muster. Denn auch bei guter Ausleuchtung haben dreidimensionale Muster immer Schatten, die das Neocognitron als Deformation der Originalmuster erkennt. Das Neocognitron verfügt zu Erkennung der Muster nur über eine Kamera und verarbeitet deswegen dreidimensionale Muster auch nur zweidimensional. Der Vollständigkeit halber werden auch noch die Ergebnisse der zweidimensionalen Muster in Tabelle 5.6 aufgeführt.

Einstellungen

Toleranz Layer 1	5
Toleranz Layer 2	6
Toleranz Layer 3	7
Lernfaktor q	10
Anzahl der Lerndurchgänge	10
Init. Der a-weights	Null
Kontrast	200
Objekte	A – Quadrat B – Rechteck C – Kreis D – Dreieck

Lernen ohne Lehrer

Lernobjekt	Erkennung			
	A	B	C	D
A	0.9910	0.9888	-	-
B	-	0.9688	-	-
C	-	-	0.9872	-
D	-	-	-	0.9877

Lernen mit Lehrer

Lernobjekt	Erkennung			
	A	B	C	D
A	0.9944	-	-	-
B	-	0.9874	-	-
C	-	-	0.9919	-
D	-	-	-	0.9967

Lernen mit Lehrer im ersten Layer

Lernobjekt	Erkennung			
	A	B	C	D
A	0.9925	-	-	-
B	-	0.9691	-	-
C	-	-	0.9853	-
D	-	-	-	0.9902

Tabelle 5.6 Ergebnisse der drei Trainingsmethoden bei zweidimensionalen Mustern

Wie nicht anders zu erwarten, sind die Ergebnisse fast identisch mit denen für dreidimensionale

Muster (Kapitel 5.4), außer beim „Lernen ohne Lehrer“. Zweidimensionale Muster sind auch einfacher für das Neocognitron zu verarbeiten, da es beim Anlernen und Erkennen keine Deformation durch Schatten geben kann.

5.6 Ergebnisse : Translation und Deformation

Das besondere am Neocognitron ist die Erkennung von sowohl deformierten als auch von verschobenen Mustern. Die Tabelle 5.7 zeigt zwei verschiedene Arten der Deformation: Die Rotation der angelerten Muster von 1° bis 5° nach links und die Deformation einer Ecke oder eines kleinen Ausschnitts beim Kreis. Es zeigt sich, daß bei der Rotation einige Muster besser erkannt werden als andere. Hingegen werden bei der anderen Deformationsart alle Muster erkannt. Der Test zur Verschiebung von Mustern in Tabelle 5.7 zeigt, daß auch hier alle Muster erkannt werden.

Einstellungen

Toleranz Layer 1	3
Toleranz Layer 2	4
Toleranz Layer 3	5
Lernfaktor Q	10000
Anzahl der Lerndurchgänge	100
Init. Der a-weights	Null
Kontrast	209
Lernverfahren	Lernen ohne Lehrer
Objekte	A – Quadrat B – Rechteck C – Kreis D – Dreieck
Position	1 – oben links 2 – oben rechts 3 – unten links 4 – unten rechts 5 – mitte

Translation

Lernobjekt	Erkennung				
	1	2	3	4	5
A	0.9922	0.9784	0.9509	0.9913	0.9965
B	0.9966	0.9978	0.9916	0.9966	0.9988
C	0.9910	0.9852	0.9798	0.9915	0.9810
D	0.9915	0.9947	0.9870	0.9953	0.9949

Rotation

Position	1 – unverändert 2 – 1° links 3 – 2° links 4 – 3° links 5 – 5° links
----------	---

Lernobjekt	Erkennung				
	1	2	3	4	5
A	0.9932	0.9848	-	-	-
B	0.9963	0.9962	0.9962	0.9915	-
C	-	-	-	-	-
D	0.9947	0.9940	0.9922	0.9888	-

Deformation

Deformationsart	1 – unverändert 2 – schräge Ecke 3 – wellige Ecke
-----------------	---

Lernobjekt	Erkennung		
	1	2	3
A	0.9951	0.9955	0.9939
B	0.9920	0.9939	0.9760
C	0.9938	0.9940	0.9949
D	0.9929	0.9918	0.6011

Tabelle 5.7 Ergebnisse von Translation und Deformation

6 Ausblick

Das Neocognitron bietet viele Möglichkeiten zur Optimierung. Eine Parallelisierung des Neocognitrons würde eine erhebliche Steigerung der Geschwindigkeit bewirken. Auch eine optimale Einstellung der Parameter würde die Ergebnisse eines Neocognitrons verbessern. Das State Key Laboratory of Software Engineering der Wuhan University hat versucht, diese Optimierung mit genetischen Algorithmen zu erreichen und ist bei einem kleinen Neocognitron zu guten Ergebnissen gelangt. Insgesamt erscheint das Potential des Neocognitrons noch nicht ausgereizt.

7 Zusammenfassung

Das Neocognitron zu Handschriftenerkennung von Fukushima ist ein hochspezialisiertes Neocognitron. Die gesamte Architektur ist auf das Erkennen von Zahlen ausgelegt und erzeugt dabei sehr gute Ergebnisse. Diese spezielle Ausrichtung hat jedoch zur Folge, daß Muster, die von diesen Zahlen abweichen, nur schlecht verarbeitet werden.

In dieser Arbeit konnte gezeigt werden, daß ein Neocognitron bei geeignetem Aufbau auch zur Identifizierung von allgemeinen Mustern geeignet ist. Hierzu war es notwendig, den von Fukushima verwendeten Aufbau des Neocognitron zu modifizieren. Der Aufbau des Neocognitrons von Fukushima faßt „einfache Merkmale“ in den S-planes mit den C-planes zusammen um so „komplexeren Merkmale“ zu erzeugen. Dieses ist aber nur möglich, wenn die Muster bekannt sind. Nur so ist eine sinnvolle Zusammenfassung von „einfachen Merkmalen“ zu „komplexeren Merkmalen“ möglich. Durch die vorgenommene Modifikation werden die „Merkmale“ nicht gezielt in bestimmten C-planes zusammengefaßt, sondern durch die Verringerung der Anzahl der S- und C-planes in jedem Layer.

Es konnte durch Versuche mit zwei- und dreidimensionalen Mustern demonstriert werden, daß infolge der vorgenommenen Modifizierung des Neocognitron-Aufbaus die Mustererkennung wesentlich flexibler funktioniert: Das Neocognitron konnte eine Reihe unterschiedlicher Muster eindeutig identifizieren.

Die sonstigen Eigenschaften des Neocognitrons verschlechterten sich dabei nicht gegenüber dem hochspezialisierten Neocognitron von Fukushima: Es verfügt immer noch über die Eigenschaft, deformierte und verschobene Muster zu erkennen.

Zusätzlich wurde gezeigt, daß durch eine geschickte Wahl des Lernfaktors q die Anzahl der Durchgänge begrenzt und somit eine Zeitersparnis erreicht werden kann.

8 Anhang A: Die Programmbedienung

Im Rahmen dieser Diplomarbeit wurde ein Programm erstellt, um das Neocoqnitron austesten zu können. Dieses Programm wurde in Delphi 7 Professional der Firma Borland geschrieben. Zur Videobearbeitung wurde die Freeware-Komponente [tscap32] verwendet.

8.1 Hardwareanforderungen

Um das Programm verwenden zu können, wird ein PC mit mindestens 1,5 Ghz und 256MB Arbeitsspeicher benötigt. Desweiteren wird eine Videokamera bzw. Webcam benötigt, welche einen Video-for-Windows-Treiber mitbringt. Wir verwenden einen Pentium 4 mit 2GHz der Firma Fujitsu Siemens und eine Webcam des Typs „PC-Cam 300“ der Firma Creative Labs.

8.2 Programmübersicht

Das Programm ist in mehrere Module aufgeteilt. Wenn man das Programm startet sieht man folgendes Bild:

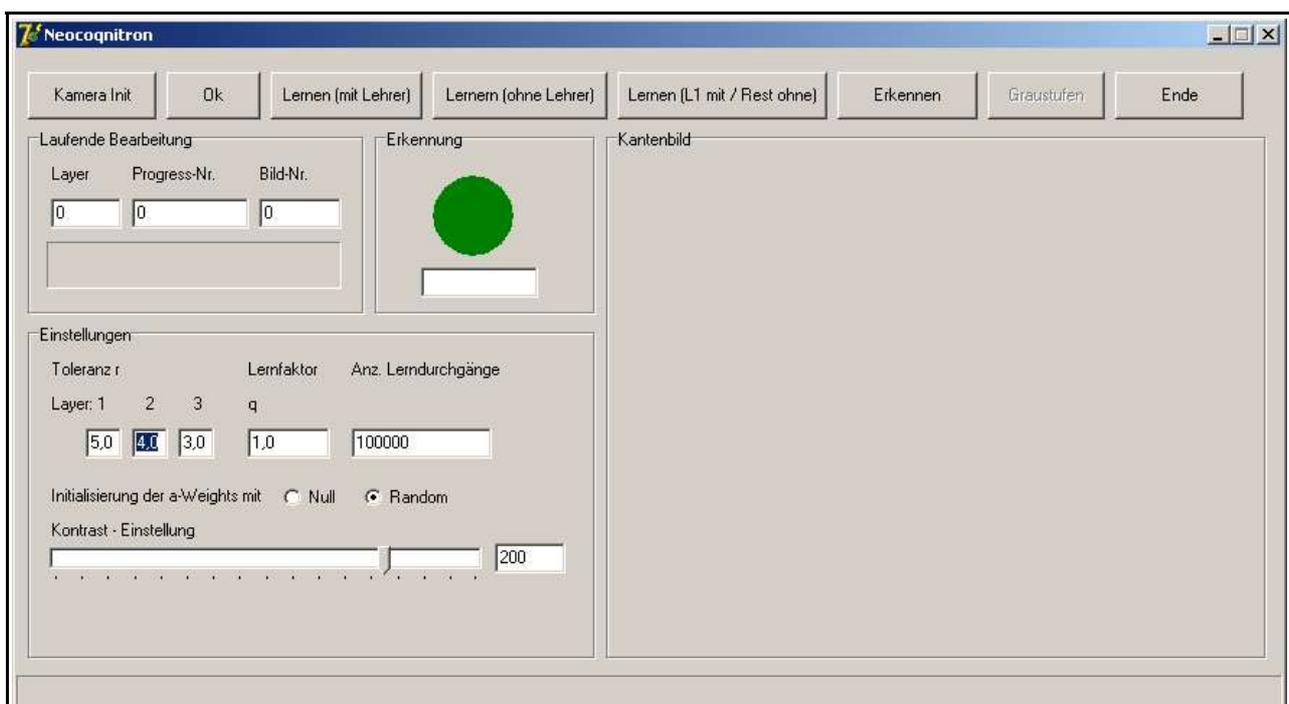
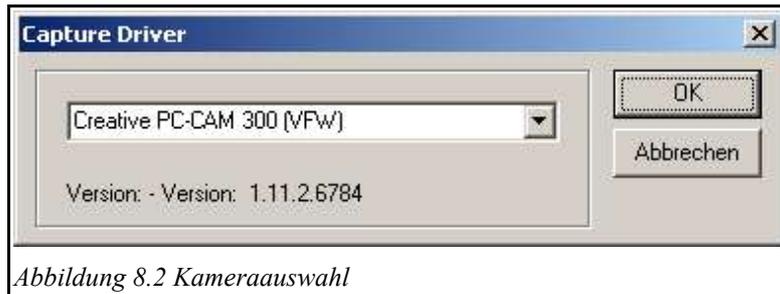


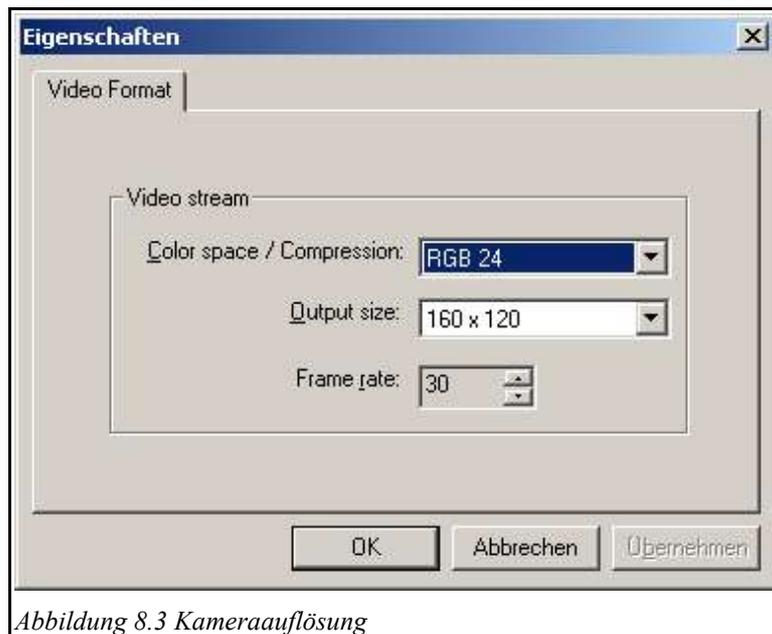
Abbildung 8.1 Hauptfenster

8.2.1 Kamera Initialisierung

Als erstes muß die Kamera initialisiert werden. Dazu klickt man auf die Schaltfläche „Kamera Init“. In dem folgenden Fenster wählt man die zu verwendende Kamera aus.



Danach wählt man die Kameraauflösung.



Das Programm ist so ausgelegt, daß jede beliebige Auflösung gewählt werden kann. Die minimale Auflösung beträgt allerdings 160 x 120 Pixel bei 24 Bit Farbtiefe.

Zum Schluß muß noch die Videokomprimierung ausgewählt werden.



Die von uns verwendete Videokomponente erlaubt aber keine Videokomprimierung. Deshalb muß man „Volle Einzelbilder“ auswählen.

8.2.2 Lernverfahren

Bevor ein Objekt angelernt werden kann, müssen noch einige Einstellungen vorgenommen werden. Zum einen müssen die Toleranzen für die einzelnen Layer eingestellt werden. Diese Toleranzen können zwischen 1 und 10 betragen, wobei niedrige Zahlen höhere Toleranzen bedeuten. Dann müssen noch der Lernfaktor und die Anzahl der Lerndurchgänge eingestellt werden. Jetzt kann man noch auswählen, ob die a-Gewichte mit Null oder mit Zufallswerten initialisiert werden. Mit der Kontrast-Einstellung kann man nach Auswahl eines Lernverfahrens bzw. der Erkennung den Kontrast der Kantendetektion einstellen.

In dem Teilfenster „Laufende Bearbeitung“ kann während der Lernphase verfolgt werden, welcher Layer gerade angelernt wird. Die Progress-Nr. ist nur eine laufende Zahl, ähnlich wie der Fortschrittsbalken, um anzuzeigen, daß die Bearbeitung läuft. Die Bildnummer wiederum zeigt nur an, daß ein Lernverfahren gestartet wurde.

8.2.3 Lernen ohne Lehrer

Nach dem man auf die Schaltfläche „Lernen (ohne Lehrer)“ geklickt hat, legt man ein Objekt vor die Kamera, richtet es aus und klickt auf die Schaltfläche „Ok“.

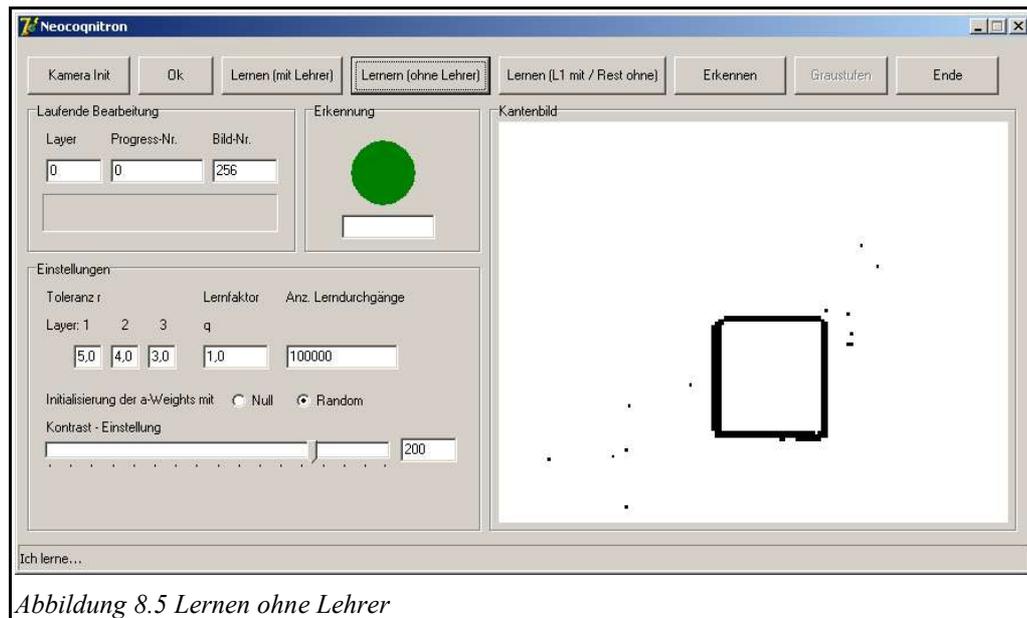


Abbildung 8.5 Lernen ohne Lehrer

Das Programm ermittelt nun selbstständig die einzelnen Muster, welche angelernt werden.

8.2.4 Lernen mit Lehrer

Wenn man „Lernen (mit Lehrer)“ auswählt, muß man die einzelnen Muster selber auswählen.

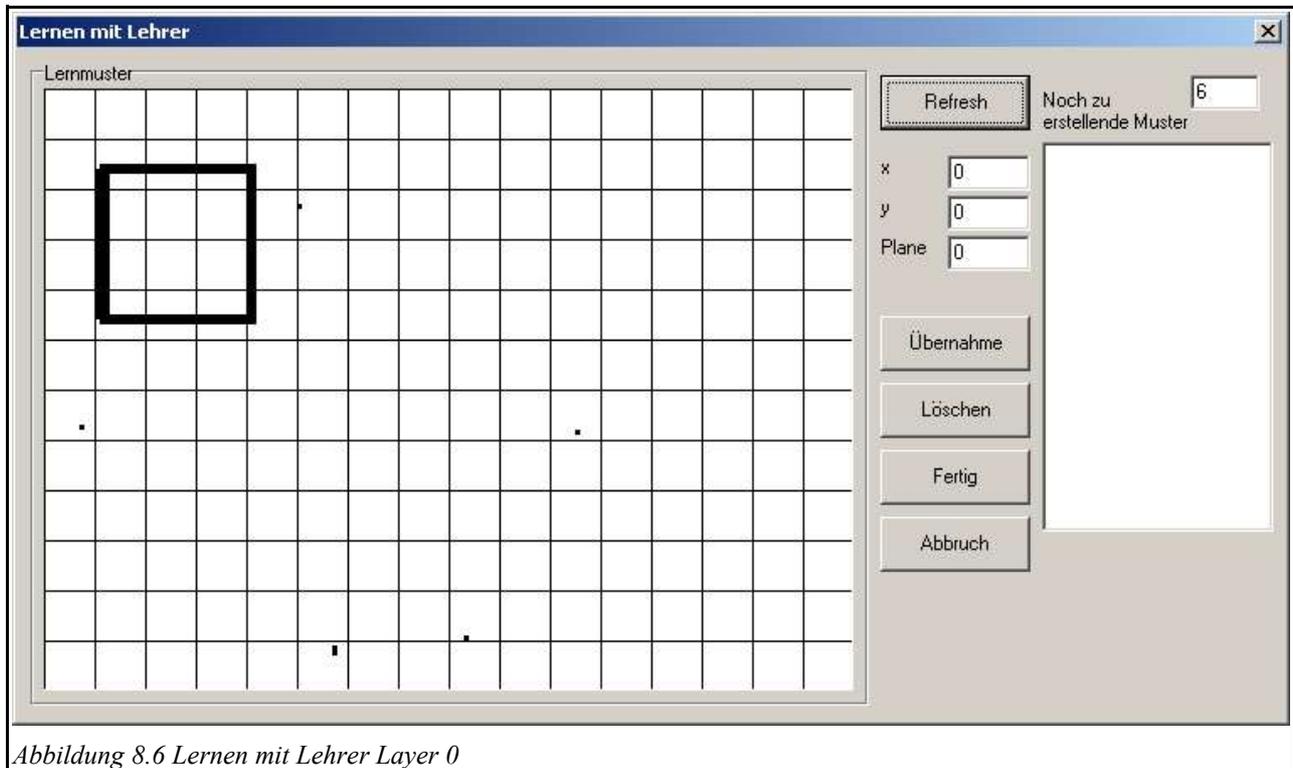


Abbildung 8.6 Lernen mit Lehrer Layer 0

Zur Erleichterung der Koordinatenermittlung der Lernmuster wurde ein Gitternetz mit einem Raster von 10 x 10 Pixel über das Bild gelegt. Mit „Refresh“ wird das Lernmuster hervorgehoben. Mit „Übernahme“ werden die Koordinaten übernommen. In der oberen rechten Ecke kann man sehen, wieviele Muster man noch auswählen muß. Mit „Löschen“ kann man Koordinaten aus der Liste wieder löschen. Wenn alle Koordinaten der zu lernenden Muster eingegeben wurden, klickt man auf „Fertig“ und die a-Gewichte für den ersten Layer werden angelernt.

Danach wird das Fenster für den zweiten Layer angezeigt.

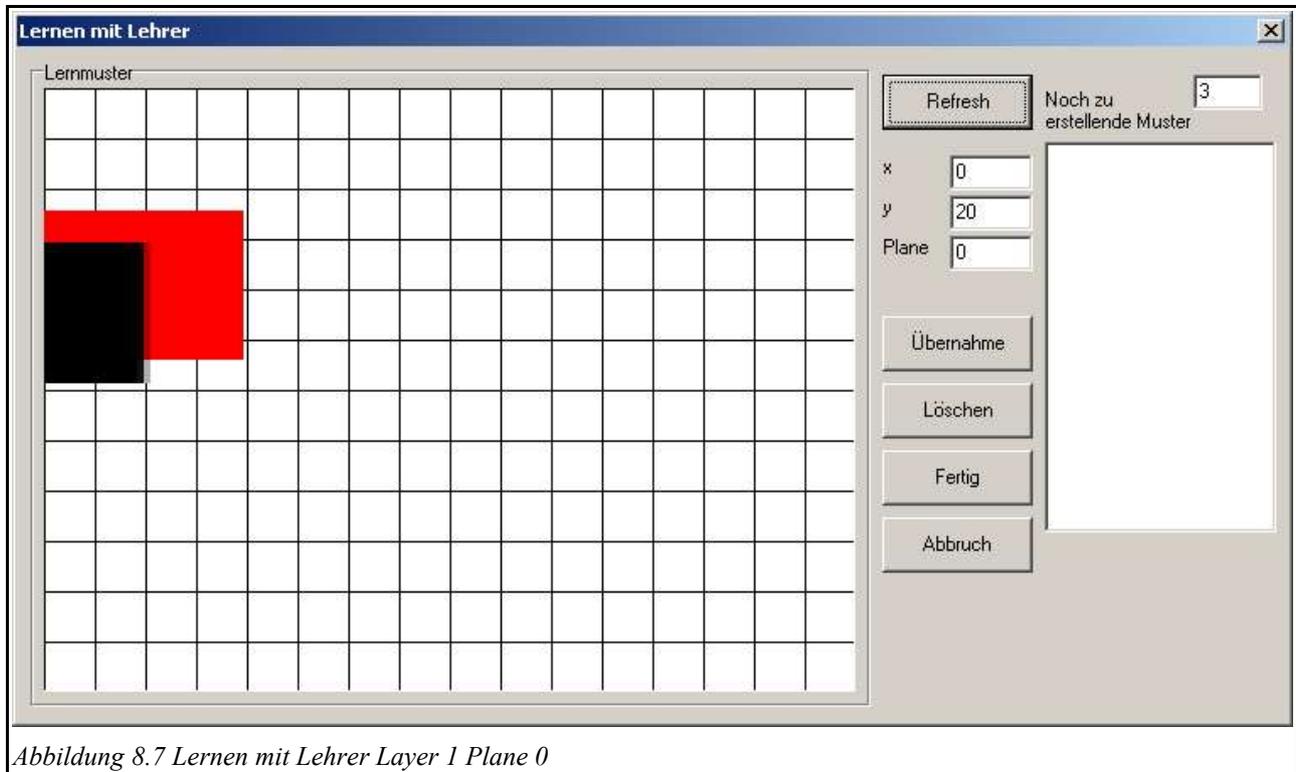


Abbildung 8.7 Lernen mit Lehrer Layer 1 Plane 0

Im zweiten Layer muß man bei den Koordinaten der Lernmuster auch die drei Planes berücksichtigen. Daher muß man mehrfach auf „Refresh“ klicken, um zu überprüfen, ob die Koordinaten korrekt sind.

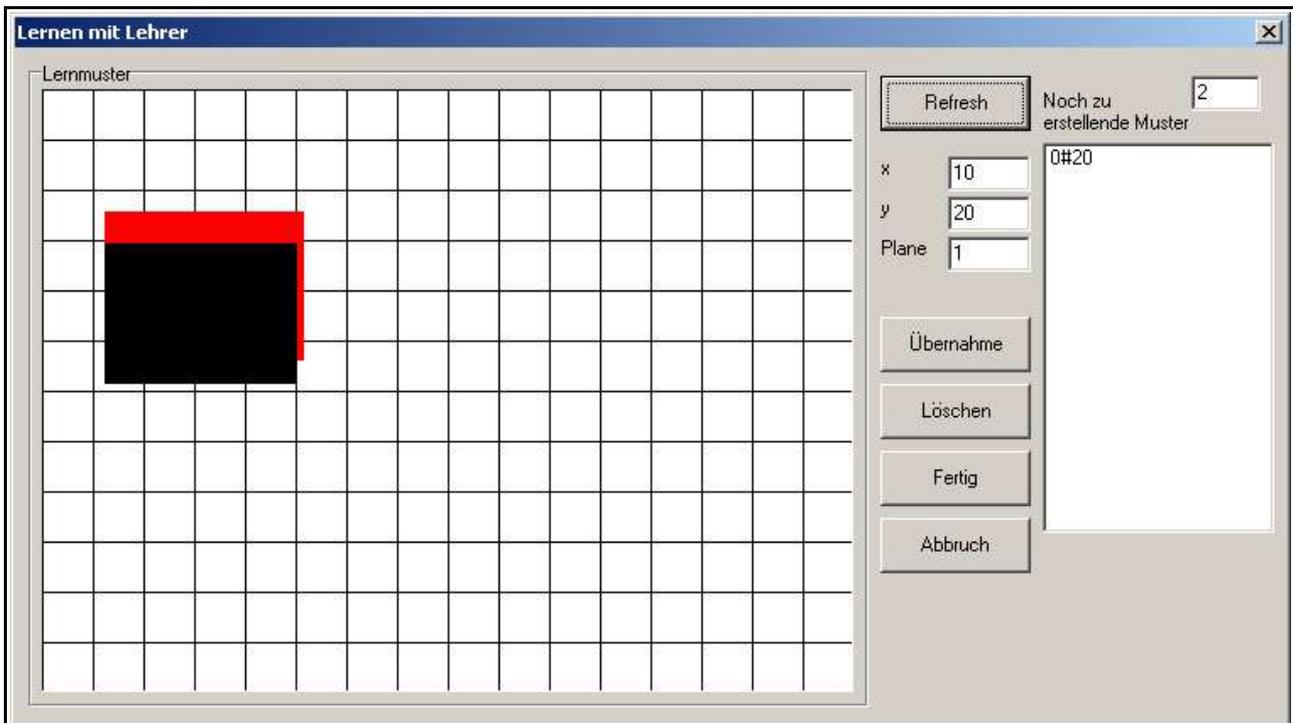


Abbildung 8.8 Lernen mit Lehrer Layer 1 Plane 1

Hier noch das Bild des dritten Planes.

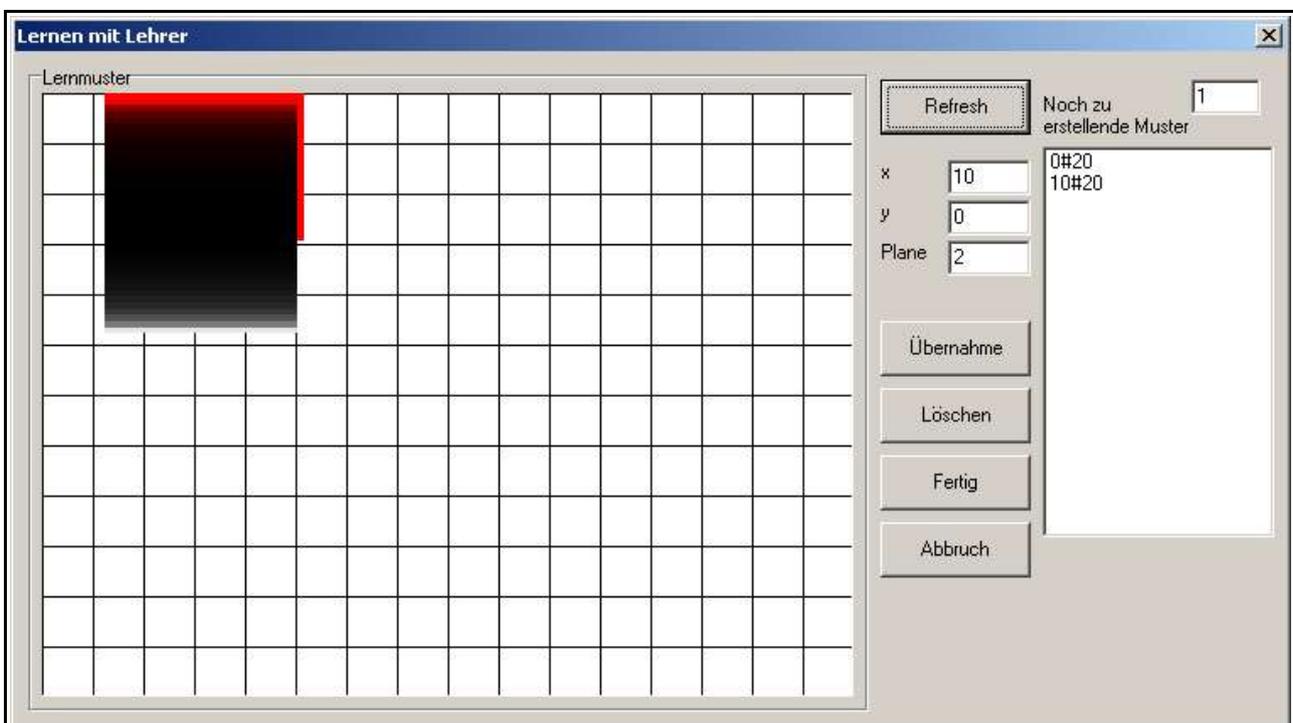


Abbildung 8.9 Lernen mit Lehrer Layer 1 Plane 2

Im dritten Layer braucht man nur noch ein Muster auszuwählen.

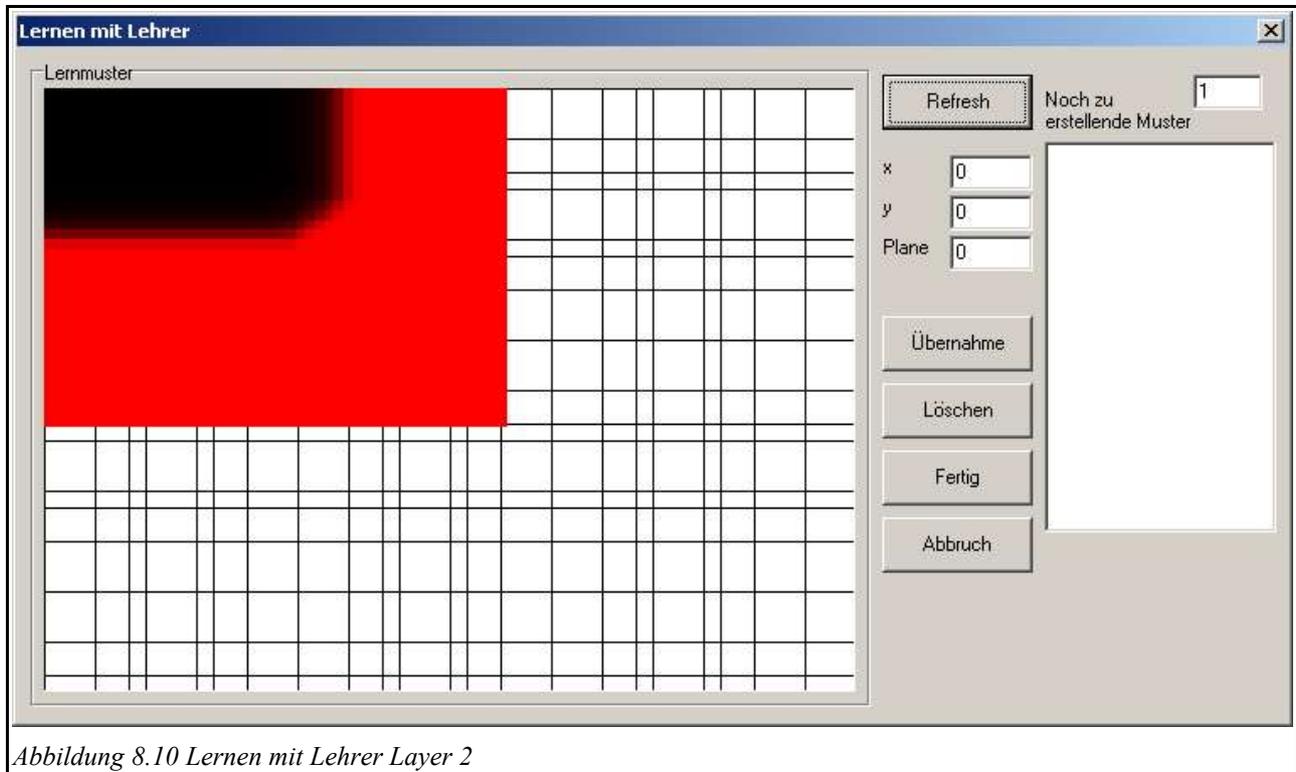


Abbildung 8.10 Lernen mit Lehrer Layer 2

8.2.5 Lernen (Layer 1 mit Lehrer / Rest ohne Lehrer)

Bei dieser Lernmethode sucht man sich nur für den ersten Layer die Muster aus. Für die Layer zwei und drei werden die Muster von dem Programm ausgewählt. Das funktioniert genauso wie beim Lernen mit Lehrer.

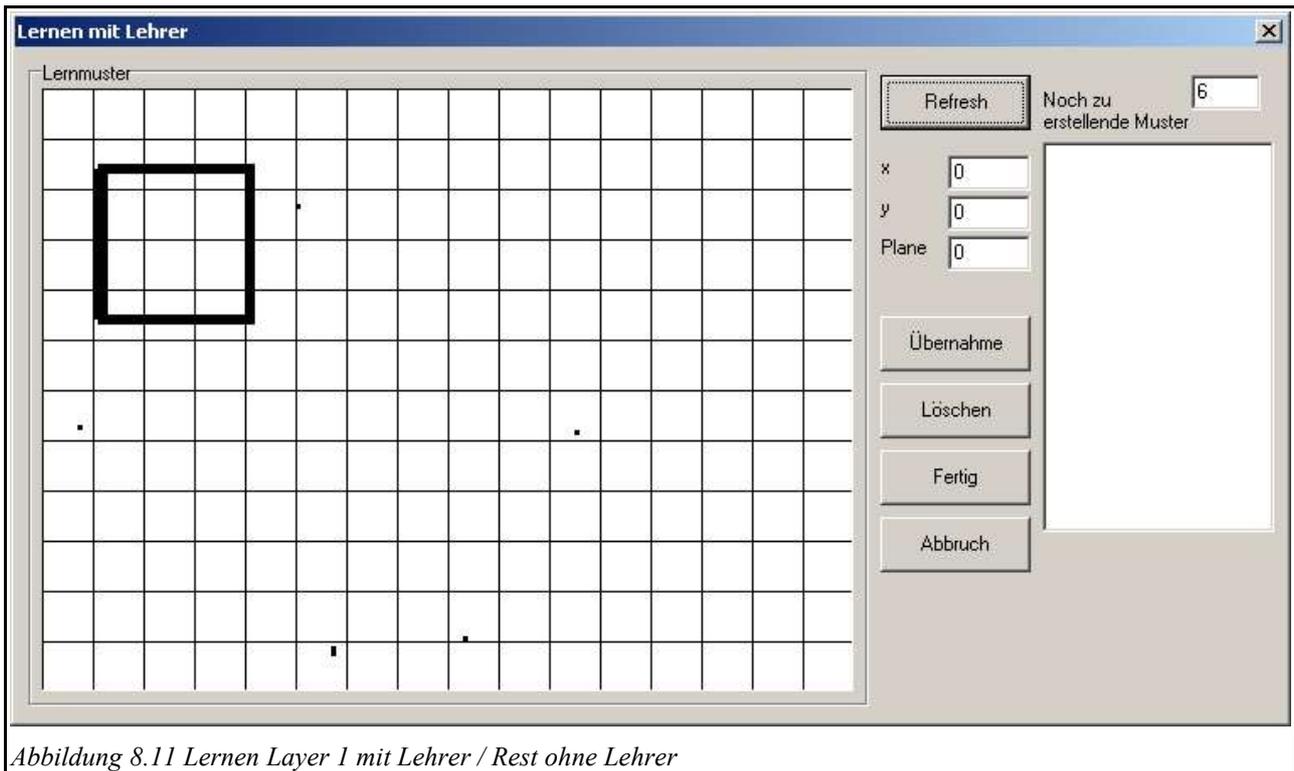
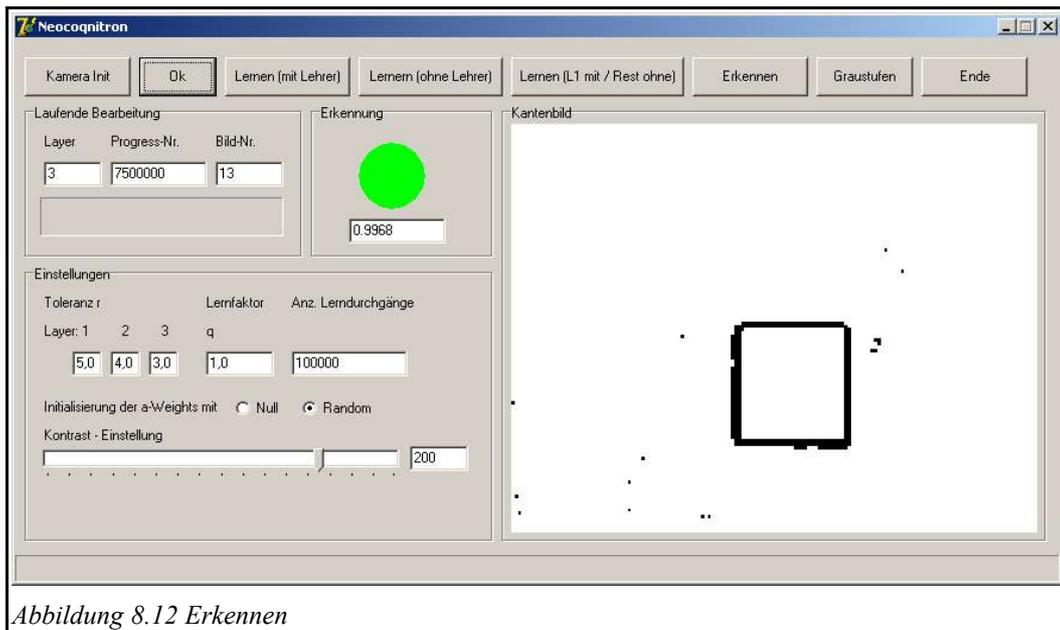


Abbildung 8.11 Lernen Layer 1 mit Lehrer / Rest ohne Lehrer

8.2.6 Erkennen

Nach dem man ein Objekt mit einer beliebigen Methode angelernt hat, kann man nun auf „Erkennen“ klicken. Nach Ausrichtung des Objekts und klicken auf „Ok“ versucht nun das Programm das angelernte Objekt zu erkennen.



Wenn das Objekt erkannt wurde, leuchtet die Lampe in hellgrün, bei Nichterkennung in rot. Der Zahlenwert unterhalb der Lampe zeigt den Grad der Erkennung an von 0.000 (nicht erkannt) bis 1.000 (voll erkannt).

8.2.7 Graustufen

Nach der Erkennung kann man sich nun die Planes der einzelnen Layer anzeigen lassen in dem man auf die Schaltfläche „Graustufen“ klickt.

Es folgt eine Serie von Graustufenbildern von der Erkennung eines Quadrates.

Auf diesen zwei Bildern wurde die linke untere Ecke eines Quadrates erkannt.

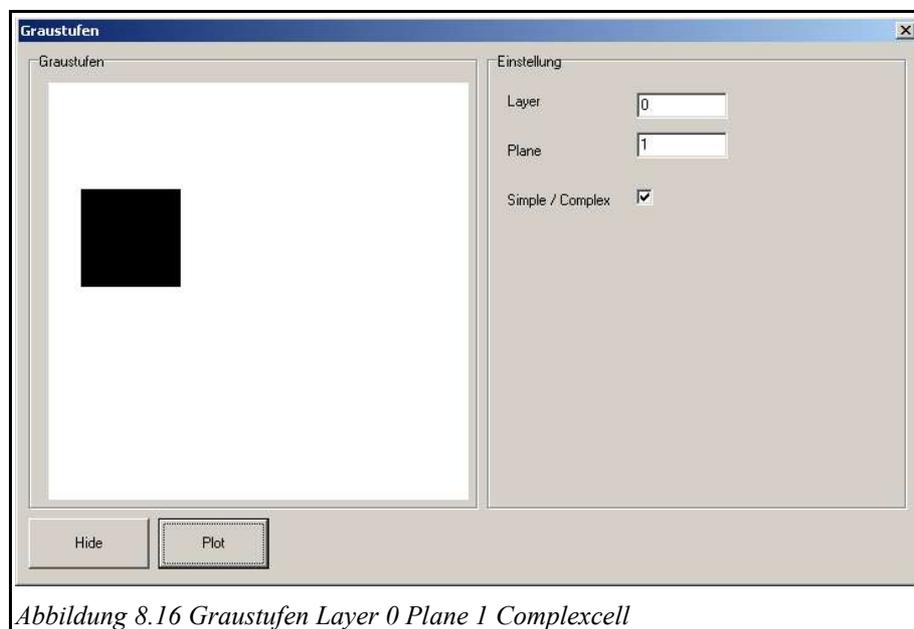
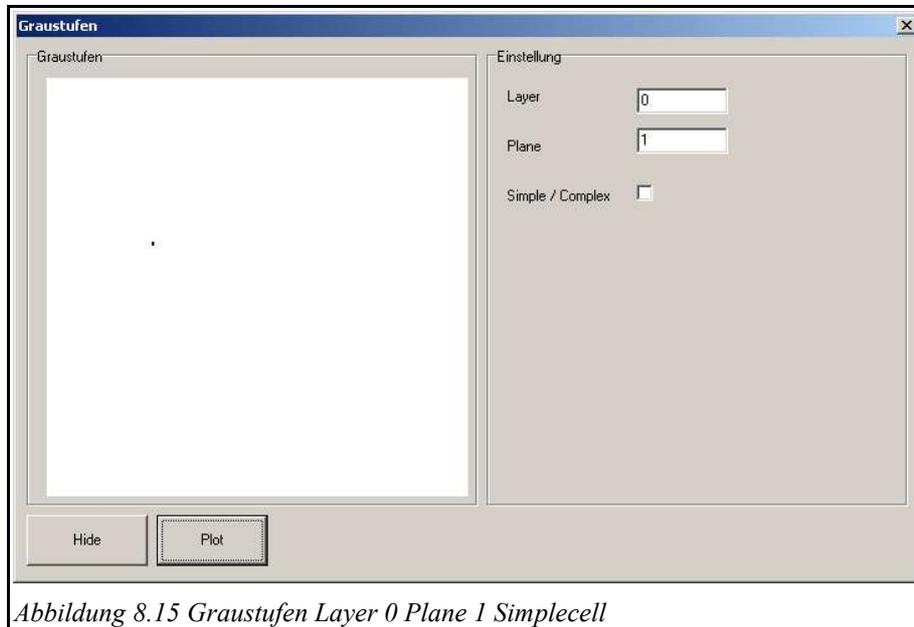


Abbildung 8.13 Graustufen Layer 0 Plane 0 Simplecell

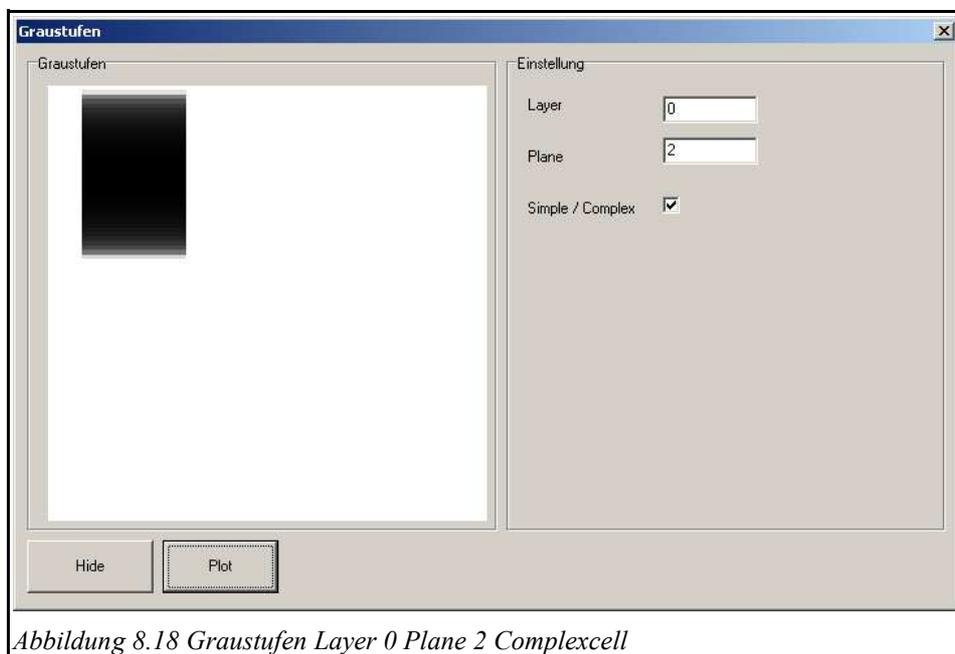
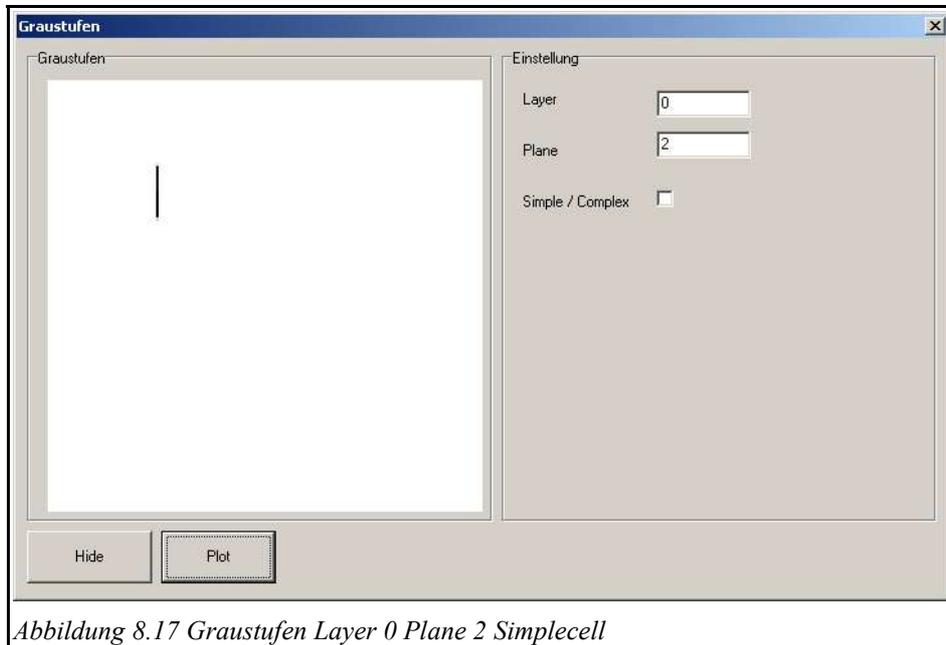


Abbildung 8.14 Graustufen Layer 0 Plane 0 Complexcell

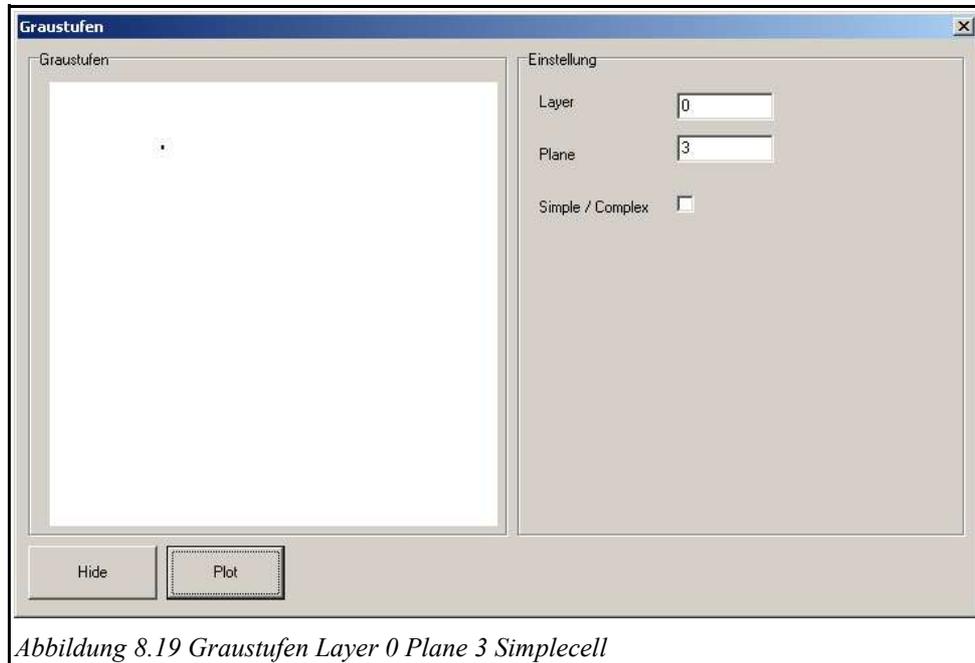
Auf diesen zwei Bildern wurde die rechte untere Ecke eines Quadrates erkannt.



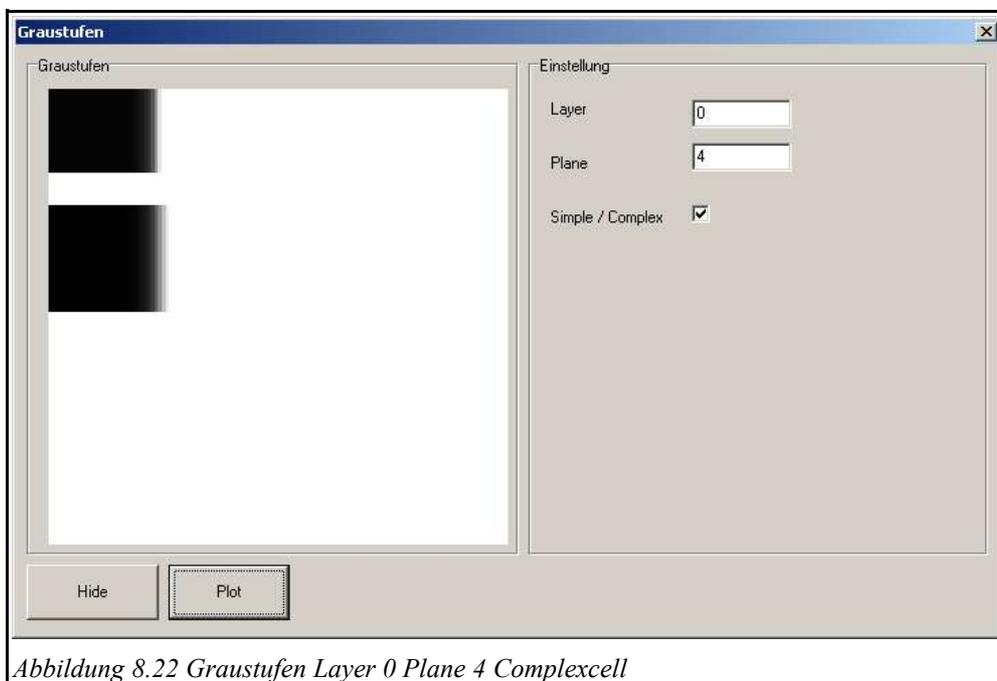
Auf diesen zwei Bildern wurde die rechte Kante eines Quadrates erkannt.



Auf diesen zwei Bildern wurde die rechte obere Ecke eines Quadrates erkannt.



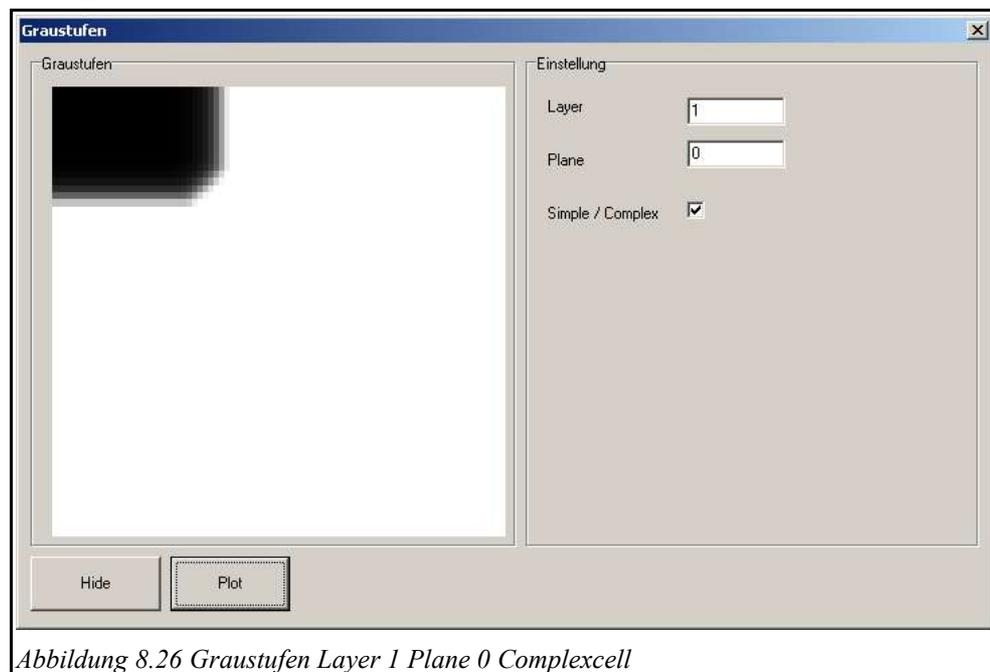
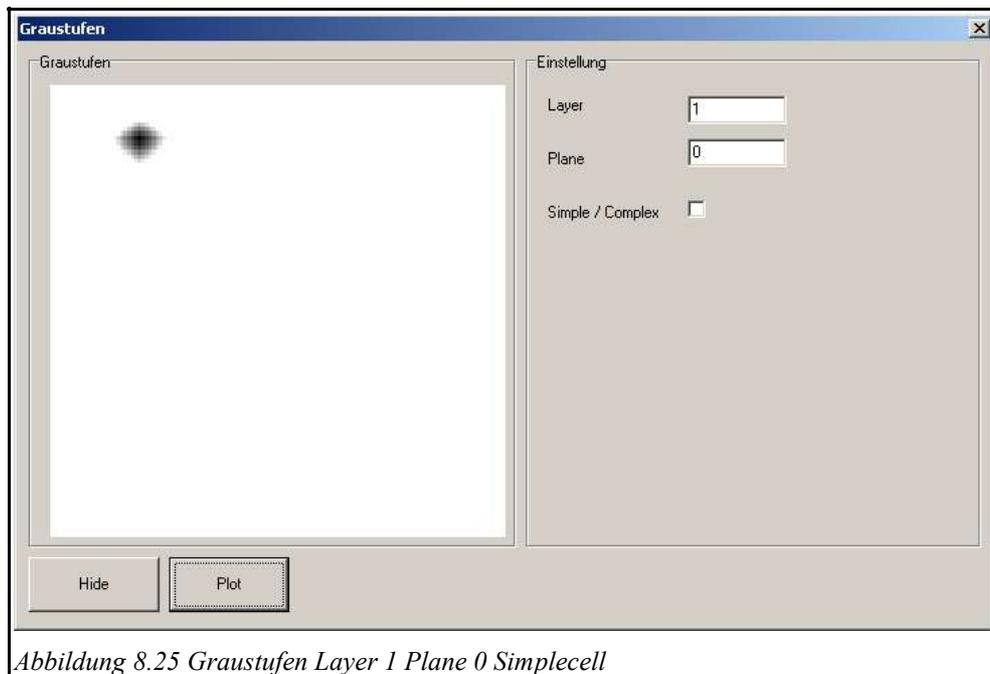
Auf diesen zwei Bildern wurden die obere und die untere Kante eines Quadrates erkannt.



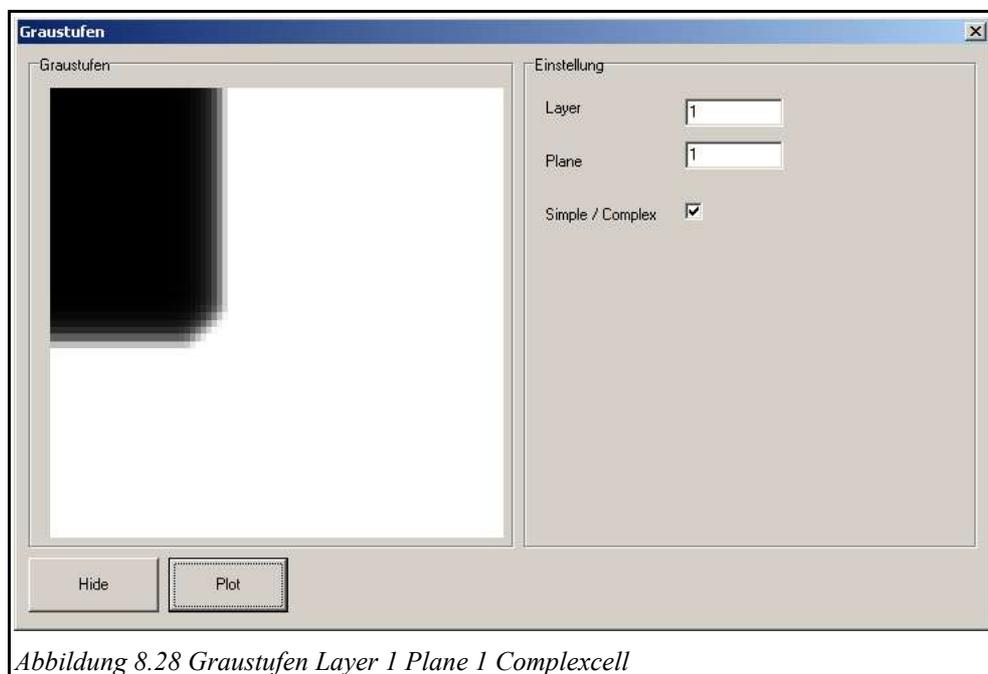
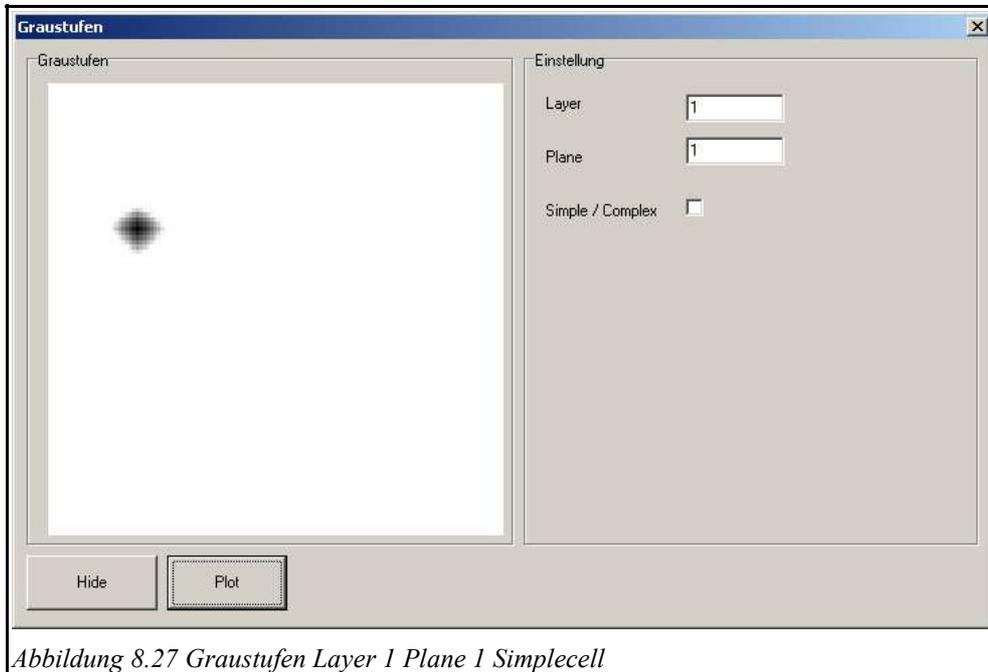
Auf diesen zwei Bildern wurde die obere linke Ecke eines Quadrates erkannt.



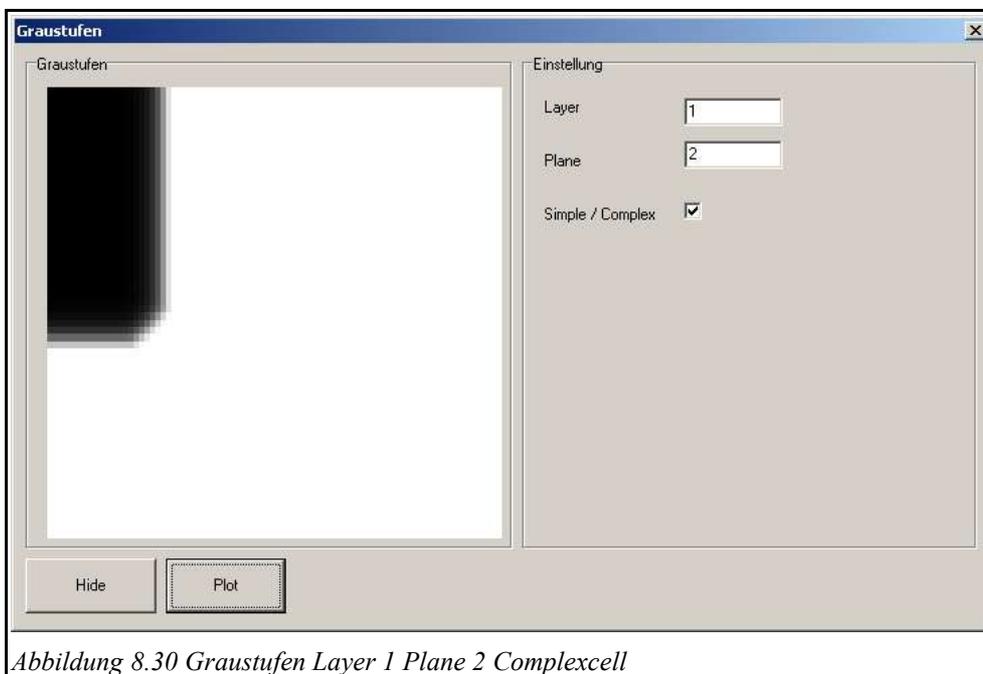
Ab dem Layer 2 kann ein Mensch nicht mehr erkennen, was der Computer sieht. Es sind Kombinationen von den Grundmustern. Hier könnte es die rechte obere Ecke in Kombination mit der oberen Kante sein.



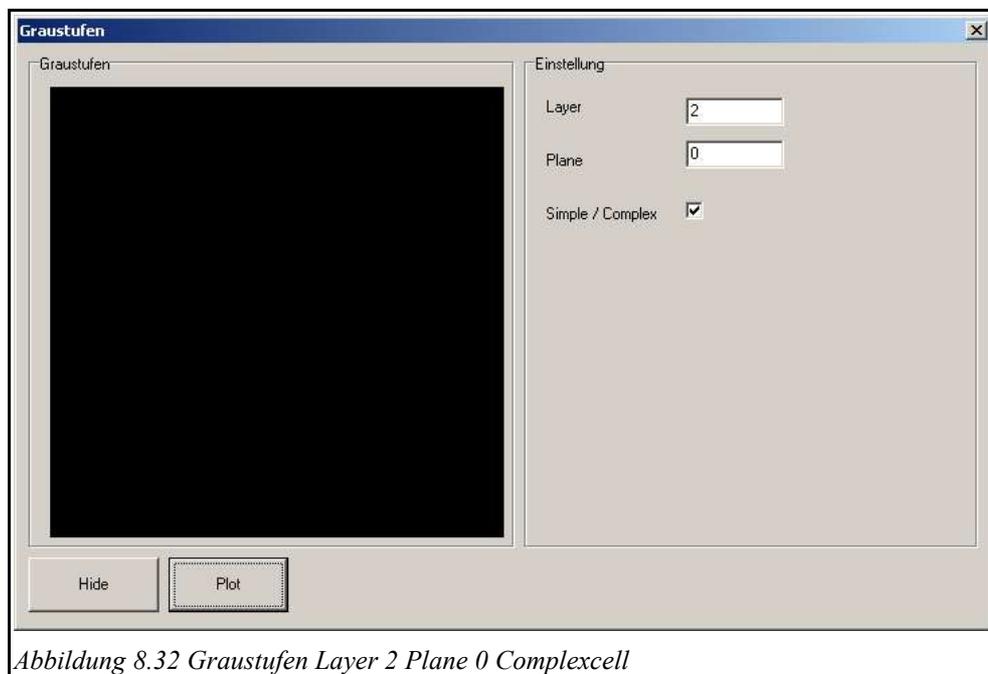
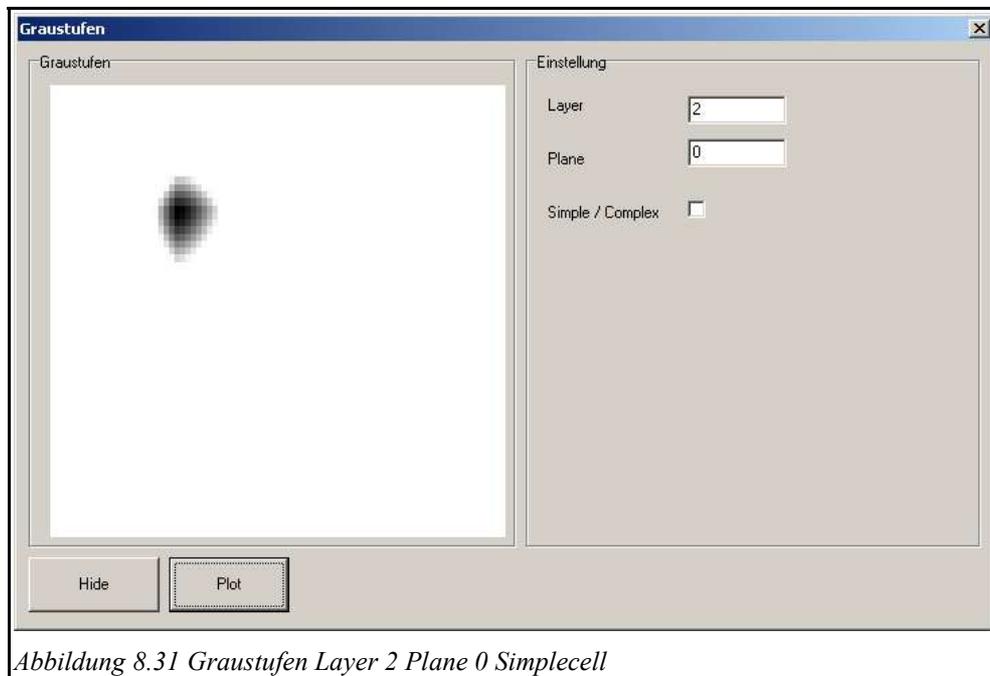
Hier könnte es die rechte untere Ecke in Kombination mit der rechten Kante sein.



Und hier ist es wahrscheinlich die linke untere Ecke in Kombination mit der linken Kante sein.



In dem dritten Layer kann ein Mensch nun wirklich nichts mehr erkennen. Man sieht nur noch einen verwaschenen Fleck.



9 Anhang B: Abbildungsverzeichnis

Abbildung 2.1 Klassische Menge M der „großen Zahlen“.....	7
Abbildung 2.2 Sprung zwischen den Mengen "groß" und "sehr groß"	8
Abbildung 2.3 Fuzzy-Menge der "großen" Zahlen.....	10
Abbildung 2.4 24-Bit Farbbild.....	11
Abbildung 2.5 Graustufenbild 8-Bit.....	12
Abbildung 2.6 Kantenbild mit lokalem Maximum.....	14
Abbildung 2.7 Kantenbild mit globalem Maximum	14
Abbildung 2.8 Kantenbild mit lokalem Maximum und Intensitätsoperator.....	15
Abbildung 2.9 "positiv" Darstellung eines Kantenbildes.....	16
Abbildung 2.10 s/w Kantenbild.....	17
Abbildung 3.1 Verbundene Neuronen	19
Abbildung 3.2 Das Auge.....	21
Abbildung 3.3 Aufbau der Retina(Netzhaut).....	22
Abbildung 3.4 Verlauf der Sehnerven.....	23
Abbildung 3.5 Aufbau von zwei überlappenden rezeptiven Feldern.....	23
Abbildung 3.6 Rezeptive Felder mit ihren off und on Gebieten.....	24
Abbildung 3.7 Hierarchische Verbindung der rezeptiven Felder mit der Simple Cell.....	25
Abbildung 3.8 Rezeptive Felder zum Erkennen von (a) helle Linien, (b) dunklen Linien, (c) hell- dunkel Kanten.....	26
Abbildung 3.9 „Complex Cell“ die über mehrere „Simple Cells“ mit rezeptiven Feldern verbunden ist	27
Abbildung 3.10 Bei Positionen a, b und c des Musters erhält die "Complex Cell" einen Reiz, bei Position d nicht	27
Abbildung 3.11 "Low-order hypercomplex cell" die mit mehreren "Complex Cells" verbunden ist	28
Abbildung 3.12 Higher-order Hypercomplex cell.....	29
Abbildung 4.1 Modell eines McCulloch-Pitts-Neuron.....	31
Abbildung 4.2 Boolesche Oder-Funktion. y folgt für I_1 und I_2 aus (3.1).....	32
Abbildung 4.3 Erster Layer s-cells US_1 mit dem Eingangsmuster UC_0	37
Abbildung 4.4 Eingangsfelder von drei S-Cells in $US_{1.6}$	38
Abbildung 4.5 Verbindung eines Teils von UC_0 über US_1 mit einer C-cell UC_1	39
Abbildung 4.6 Die Verbindungen von C-planes mit den S-planes.....	40
Abbildung 4.7 Verbindung einer S-plane mit allen C-planes des vorherigen Layers	41
Abbildung 4.8 Eine S-cell die mit mehreren Eingabefeldern verbunden ist.....	42
Abbildung 4.9 Die Verbindung von einer S-cell mit allen Eingabefeldern einer Region im vorhergehenden C-plane....	43
Abbildung 4.10 Durch die Veränderung des Eingabefeldes (hier eine Vergrößerung auf 5x5 Pixel) ist eine Verringerung der C-cells in den C-planes möglich.....	44
Abbildung 4.11 Alle Layer eines Neocognitrons zur Erkennung der Zahlen 0 bis 9 . Im letzten Layer bestehen die C-planes nur noch aus einer C-cell und korrespondieren damit direkt mit den zu erkennenden Zahlen.....	45
Abbildung 4.12 Die „Merkmale“ zum Anlernen des ersten Layers.....	48
Abbildung 4.13 Die "Merkmale" zum Anlernen des zweiten Layers. Einige "Merkmale" sind in verschiedenen Variationen vorhanden.....	49

Abbildung 4.14 Die "Merkmale" für den dritten Layer. Einige "Merkmale" kommen wieder in verschiedenen Variationen vor.....	50
Abbildung 4.15 Der komplette Satz der Muster zum Anlernen des vierten Layer. Die Zahlen kommen in verschiedenen Variationen zum Anlernen vor.	51
Abbildung 4.16 Die aktive Zelle (schwarz) hat die Position $n = (2,12)$	53
Abbildung 4.17 Die Gewichte (weights) der verschiedenen Typen von Planes.....	54
Abbildung 5.1 Aufbau eines Neocognitrons zur Erkennung eines Musters.....	58
Abbildung 8.1 Hauptfenster.....	72
Abbildung 8.2 Kameraauswahl.....	73
Abbildung 8.3 Kameraauflösung.....	73
Abbildung 8.4 Videokomprimierung.....	74
Abbildung 8.5 Lernen ohne Lehrer.....	75
Abbildung 8.6 Lernen mit Lehrer Layer 0.....	76
Abbildung 8.7 Lernen mit Lehrer Layer 1 Plane 0.....	77
Abbildung 8.8 Lernen mit Lehrer Layer 1 Plane 1.....	78
Abbildung 8.9 Lernen mit Lehrer Layer 1 Plane 2.....	78
Abbildung 8.10 Lernen mit Lehrer Layer 2.....	79
Abbildung 8.11 Lernen Layer 1 mit Lehrer / Rest ohne Lehrer.....	80
Abbildung 8.12 Erkennen.....	81
Abbildung 8.13 Graustufen Layer 0 Plane 0 Simplecell.....	82
Abbildung 8.14 Graustufen Layer 0 Plane 0 Complexcell.....	82
Abbildung 8.15 Graustufen Layer 0 Plane 1 Simplecell.....	83
Abbildung 8.16 Graustufen Layer 0 Plane 1 Complexcell.....	83
Abbildung 8.17 Graustufen Layer 0 Plane 2 Simplecell.....	84
Abbildung 8.18 Graustufen Layer 0 Plane 2 Complexcell.....	84
Abbildung 8.19 Graustufen Layer 0 Plane 3 Simplecell.....	85
Abbildung 8.20 Graustufen Layer 0 Plane 3 Complexcell.....	85
Abbildung 8.21 Graustufen Layer 0 Plane 4 Simplecell.....	86
Abbildung 8.22 Graustufen Layer 0 Plane 4 Complexcell.....	86
Abbildung 8.23 Graustufen Layer 0 Plane 5 Simplecell.....	87
Abbildung 8.24 Graustufen Layer 0 Plane 5 Complexcell.....	87
Abbildung 8.25 Graustufen Layer 1 Plane 0 Simplecell.....	88
Abbildung 8.26 Graustufen Layer 1 Plane 0 Complexcell.....	88
Abbildung 8.27 Graustufen Layer 1 Plane 1 Simplecell.....	89
Abbildung 8.28 Graustufen Layer 1 Plane 1 Complexcell.....	89
Abbildung 8.29 Graustufen Layer 1 Plane 2 Simplecell.....	90
Abbildung 8.30 Graustufen Layer 1 Plane 2 Complexcell.....	90
Abbildung 8.31 Graustufen Layer 2 Plane 0 Simplecell.....	91
Abbildung 8.32 Graustufen Layer 2 Plane 0 Complexcell.....	91

10 Anhang C: Tabellenverzeichnis

Tabelle 5.1 Auswertung von „Lernen mit Lehrer nur im ersten Layer“	61
Tabelle 5.2 "Lernen mit Lehrer im ersten Layer" bei der Erkennung der vier Muster.....	62
Tabelle 5.3 Auswertung "Lernen mit Lehrer".....	63
Tabelle 5.4 Auswertung "Lernen ohne Lehrer".....	63
Tabelle 5.5 Verschiedene Lerndurchgänge und Lernfaktorn beim "Lernen mit Lehrer"	65
Tabelle 5.6 Ergebnisse der drei Trainingsmethoden bei zweidimensionalen Mustern	67
Tabelle 5.7 Ergebnisse von Translation und Deformation	69

11 Anhang D: Literaturverzeichnis

- blco70: C. Blakemore, G. F. Cooper, Development of the brain depends on the visual environment, 1970, *Nature* 228, 477-478
- bo98: Hans-Heinrich Bothe, *Neuro-Fuzzy-Methoden Einführung in Theorie und Anwendung*, 1998, Springer-Verlag, 3-540-57966-4
- frsk91: James A. Freeman, David M. Skapura, *Neural Networks Algorithms, Applications and Programming Techniques*, 1991, Addison-Wesley Verlag, 0-201-51376-5
- fu75: K. Fukushima, Cognitron: A self-organizing multilayered neural network, 1975, *Biological Cybernetics* 20, 121-136
- hama75: P. Hammond, D. M. MacKay, Differential response of cat visual cortical cells to textured stimuli, 1975, *Experimental Brain Research*, 22pp, 427-430
- huwi59: D. H. Hubel, T. N. Wiesel, Receptive fields of single neurons in the cat's striate cortex, 1959, *Journal of Physiology*, 148pp, 574-591
- ka93: Jörg Kahlert, *Fuzzy-Logik und Fuzzy-Control*, 1993, Vieweg-Verlag, 3-528-05304-6
- ki94: W. Kinnebrock, *Neuronale Netze Grundlagen, Anwendungen, Beispiele*, 1994, Oldenbourg Verlag, 3-486-22947-8
- lo94: R. Lowel, *The Neocognitron as a system for handwritten chars recognitron*, 1994
- lu13: Jan Lukasiewicz, *Die logischen Grundlagen der Wahrscheinlichkeitsrechnung*, 1913, ,
- mepi43: W. S. McCulloch, W. Pitts, A Logical Calculus of Ideas Immanent in Nervous Activity, 1945, *Bulletin of Mathematical Biophysics* 5,
- mo75: J. A. Movshon, The velocity tuning of single units in cat striate cortex, 1975, *Journal of Physiology*, 249pp, 445-468
- ro58: F. Rosenblatt, The Perceptron: A probabilistic model for information storage, 1958, *Psychological Review* 65,
- si75: A. M. Sillito, The contribution of inhibitory mechanisms to the receptive field properties, 1975, *Journal of Physiology*, 250pp, 305-329
- st93: Dr. Rainer Steinbrecher, *Bildverarbeitung in der Praxis*, 1993, R. Oldenbourg Verlag, 3-489-22372-0
- ti97: Hamid R. Tizhoosh, *Fuzzy-Bildverarbeitung Einführung in Theorie und Praxis*, 1997, ,
- za65: Lotfi A. Zadeh, Fuzzy Sets, 1965, In *Information and Control* 8, 338-353
- ze94: Andreas Zell, *Simulation Neuronaler Netze*, 1994, Addison-Wesley Verlag, 3-89319-554-8
- tscap32: Thomas Stuefe, *Delphi Video Capture Component*, 2003, <http://www.tstech.de/tscap32>
- [4] <http://cs.felk.cvut.cz/~neurony/neocog/en/index.html>

Bilderquellen

- [1] www.merian.fr.bw.schule.de
- [2] www.ece.utep.edu
- [3] R. Lowel, *The Neocognitron as a system for handwritten chars recognitron*, 1994
- [4] <http://cs.felk.cvut.cz/~neurony/neocog/en/index.html>

Erklärung

Hiermit erkläre ich, daß der gekennzeichnete Teil der Diplomarbeit von mir selbstständig verfaßt und angefertigt wurde, nur die angegebenen Quellen und Hilfsmittel benutzt wurden und Zitate kenntlich gemacht wurden.