

Das Neocognitron

Matthias Jauernig, 02.05.07

Überblick

1. Künstliche Neuronale Netze allgemein
2. Neocognitron-Kurzüberblick
3. Biologisches Vorbild
4. Netzwerk-Struktur
5. Verbindungen und Gewichte
6. Prozess der Mustererkennung
7. Überwachtes Lernen
8. Modifikationen und Anwendungsgebiete
9. Zusammenfassung

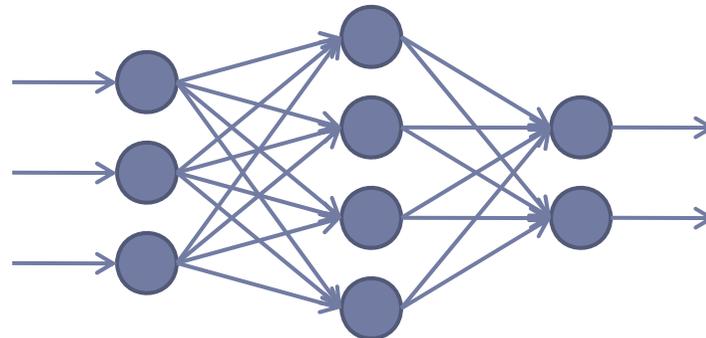


(1) Künstliche Neuronale Netze
allgemein



KNNs allgemein

- ▶ Allg.: KNN = gerichteter Graph:
 - ▶ Kanten: gewichtet → Gewichte = implizites Wissen
 - ▶ Knoten: „Neuronen“ → primitive signalverarbeitende Elemente (Prozessoren)
- ▶ Beispiel: Vorwärtsgerichtetes Netz



Knoten:

Eingabe: gewichtete Summe der Vorgänger-Ausgabe

Ausgabe: Schwellenwert-Funktion, z.B. Sigmoid-Fkt.



KNNs allgemein

- ▶ Anwendungsgebiete (Auszug):
 - ▶ Optimierung
 - ▶ Robotik
 - ▶ Zeitreihenanalyse
 - ▶ Klassifikationsaufgaben: Bilderkennung, Texterkennung, Gesichtserkennung



(2) Neocognitron-Kurzüberblick



Neocognitron-Kurzüberblick

- ▶ 1980 von Prof. Kuniyuki Fukushima vorgestellt
- ▶ Nachfolger des Cognitrons
- ▶ Modell des visuellen Wahrnehmungssystems von Säugetieren (am Beispiel von Katzen)
- ▶ Entwickelt zur Erkennung visueller Muster
- ▶ Weiteres Ziel: Vorgänge im Gehirn besser verstehen
- ▶ Eigenschaften: tolerant gegenüber Verschiebungen, Skalierungen und Deformationen
- ▶ H.-Nielsen 1990: kompliziertestes KNN seiner Zeit



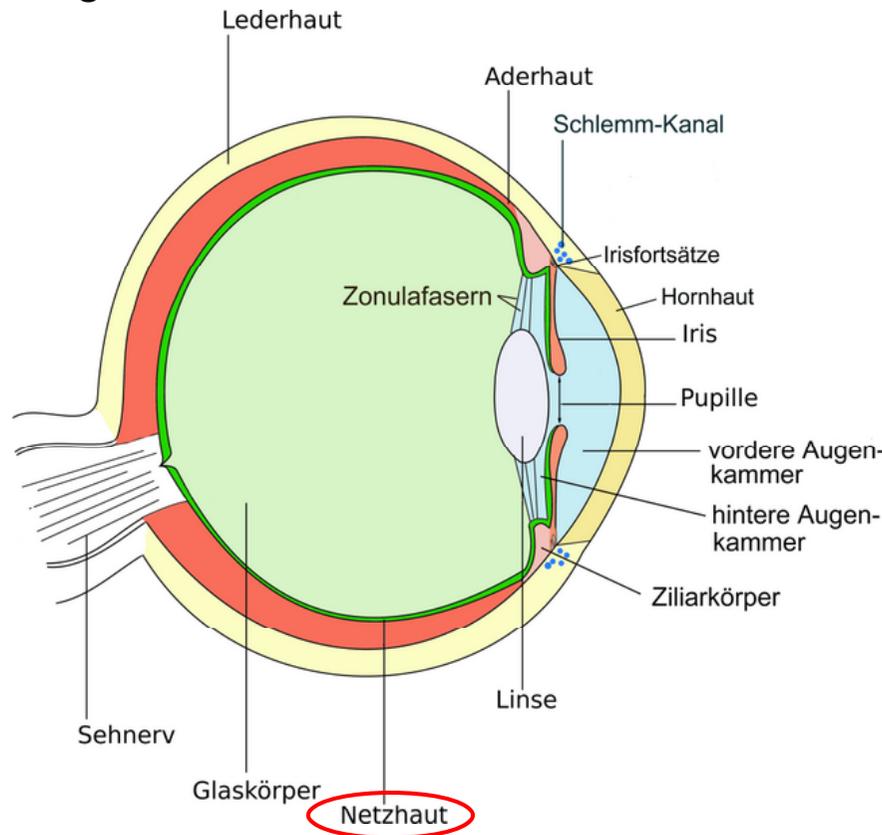
(3) Biologisches Vorbild



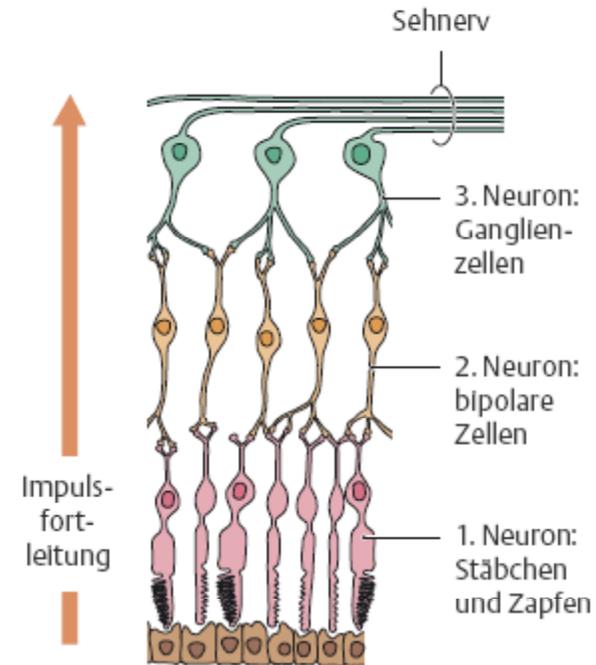
Biologisches Vorbild

► Sehsystem von Säugetieren:

Auge:



Retina (Netzhaut):



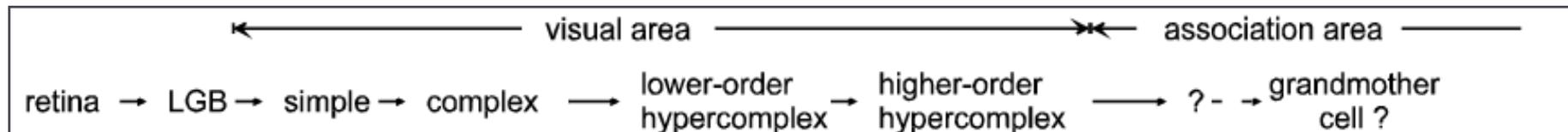
Biologisches Vorbild

▶ Modell von Hubel und Wiesel:

Untersuchung des Sehsystems von Katzen:



Modell:



- ▶ LGB = Lateral Geniculate Body – Umschaltstelle zum visuellen Kortex
- ▶ Hierarchische Merkmalsextraktion
- ▶ Klassifizierung durch Großmutter-Zellen

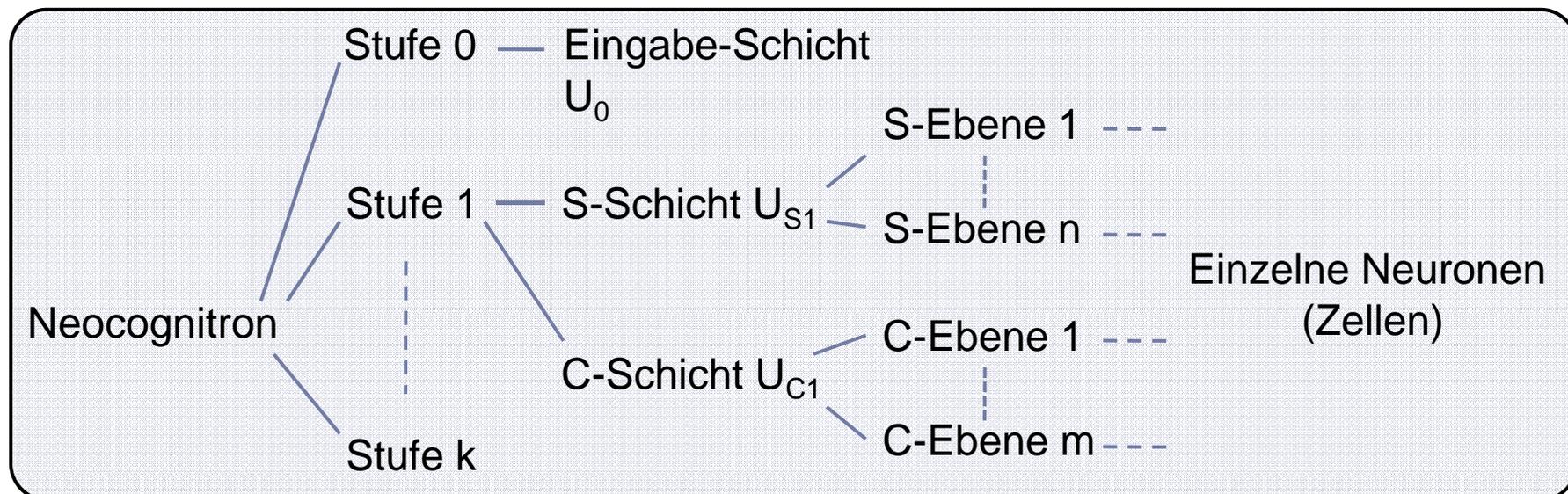


(4) Netzwerk-Struktur



Netzwerk-Struktur

- ▶ Allgemein: → als Blackbox ansehbar
 - ▶ Stufenweise Anordnung
 - ▶ Vorwärtsgerichteter Signalfluss
 - ▶ Keine vollständige Stufen-Verbindung, sondern selektiv
- ▶ Etwas konkreter:



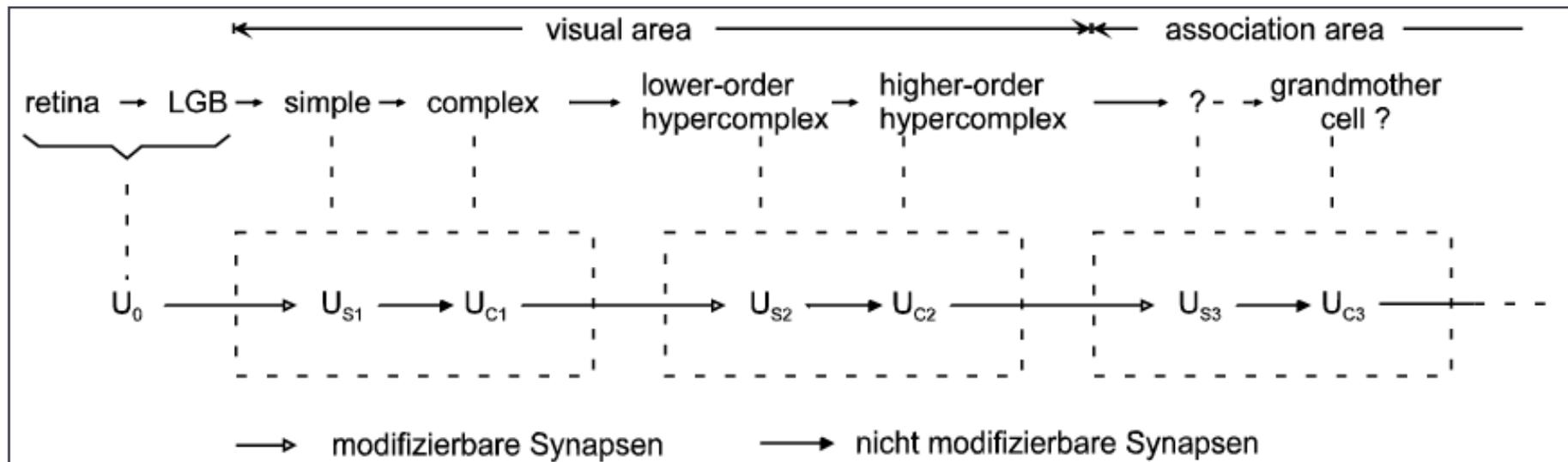
Netzwerk-Struktur

- ▶ **Eine Stufe:**
 - ▶ Extrahiert Muster bestimmter Komplexität
- ▶ **Eine Schicht:**
 - ▶ S-Schicht: Erkennt angelernte Muster
 - ▶ C-Schicht: macht Erkennung tolerant gegenüber Verschiebungen, Skalierungen und Deformationen
- ▶ **Eine Ebene:**
 - ▶ S-Ebene: ist auf bestimmtes Muster angelernt und erkennt dieses an beliebiger Position im Eingaberaum
- ▶ **Eine Zelle:**
 - ▶ Erkennt ein Muster an einer bestimmten Position



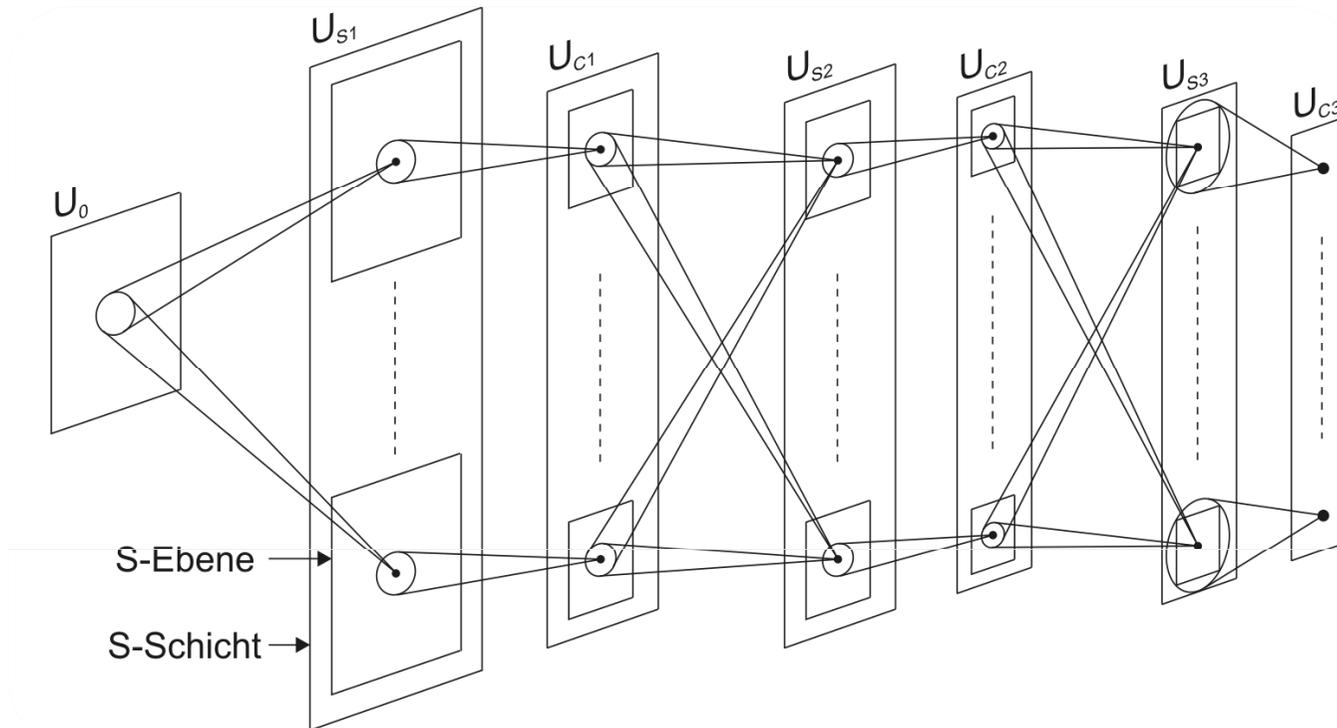
Netzwerk-Struktur

- Orientierung am biologischen Vorbild:



Merkmalsextraktion

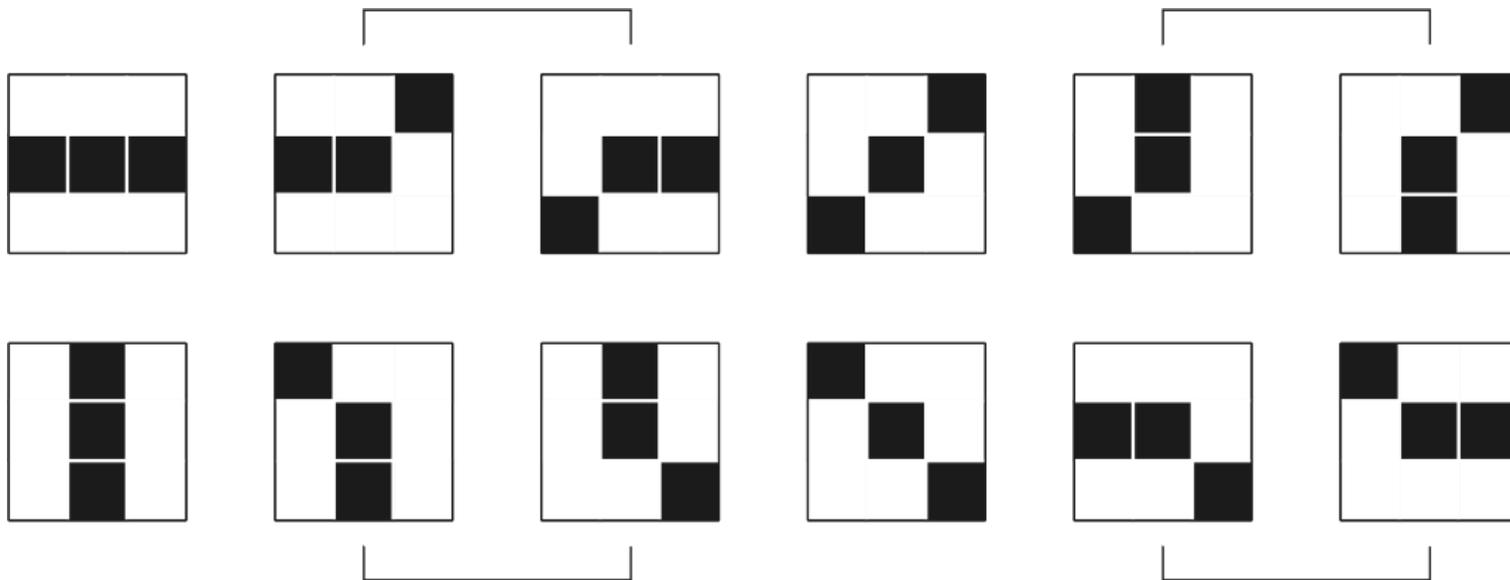
- ▶ Prinzip: hierarchische Merkmalsextraktion



Steigende Musterkomplexität

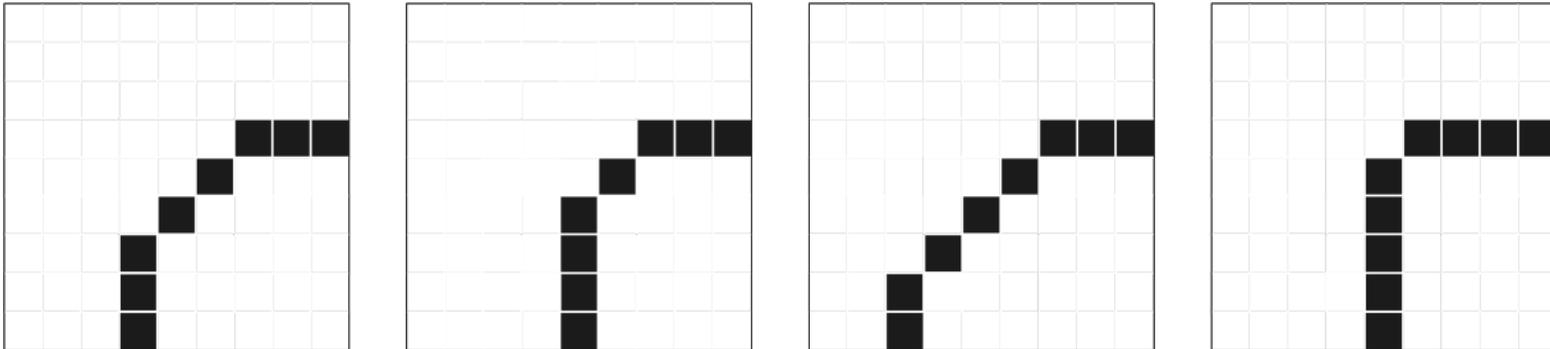
Merkmalsextraktion

- ▶ Beispiel: von U_{S_1} gelernte Muster
 - ▶ Primitive Linienformen unterschiedlicher Orientierung



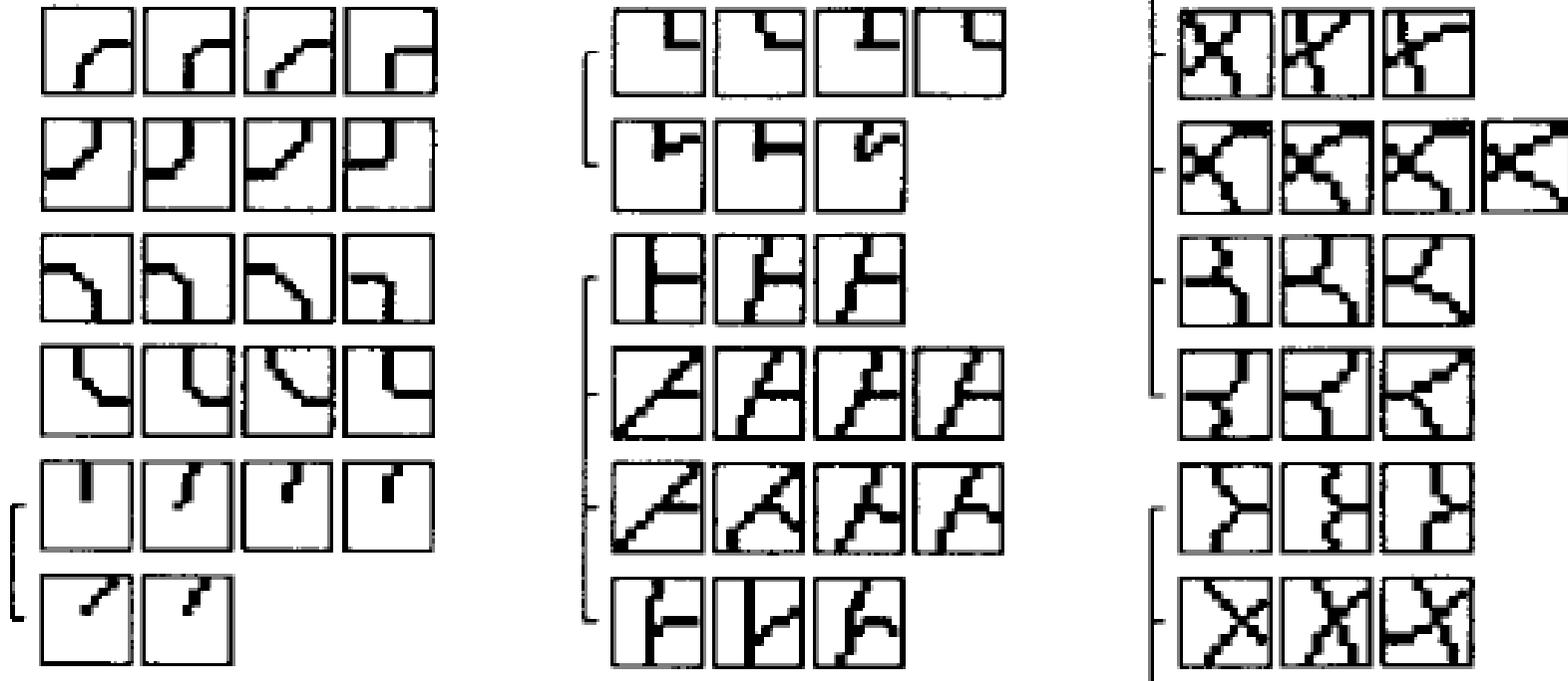
Merkmalsextraktion

- ▶ Beispiel: von U_{S_2} gelernte Muster
 - ▶ Zusammensetzung primitiver Linien der vorherigen Stufe



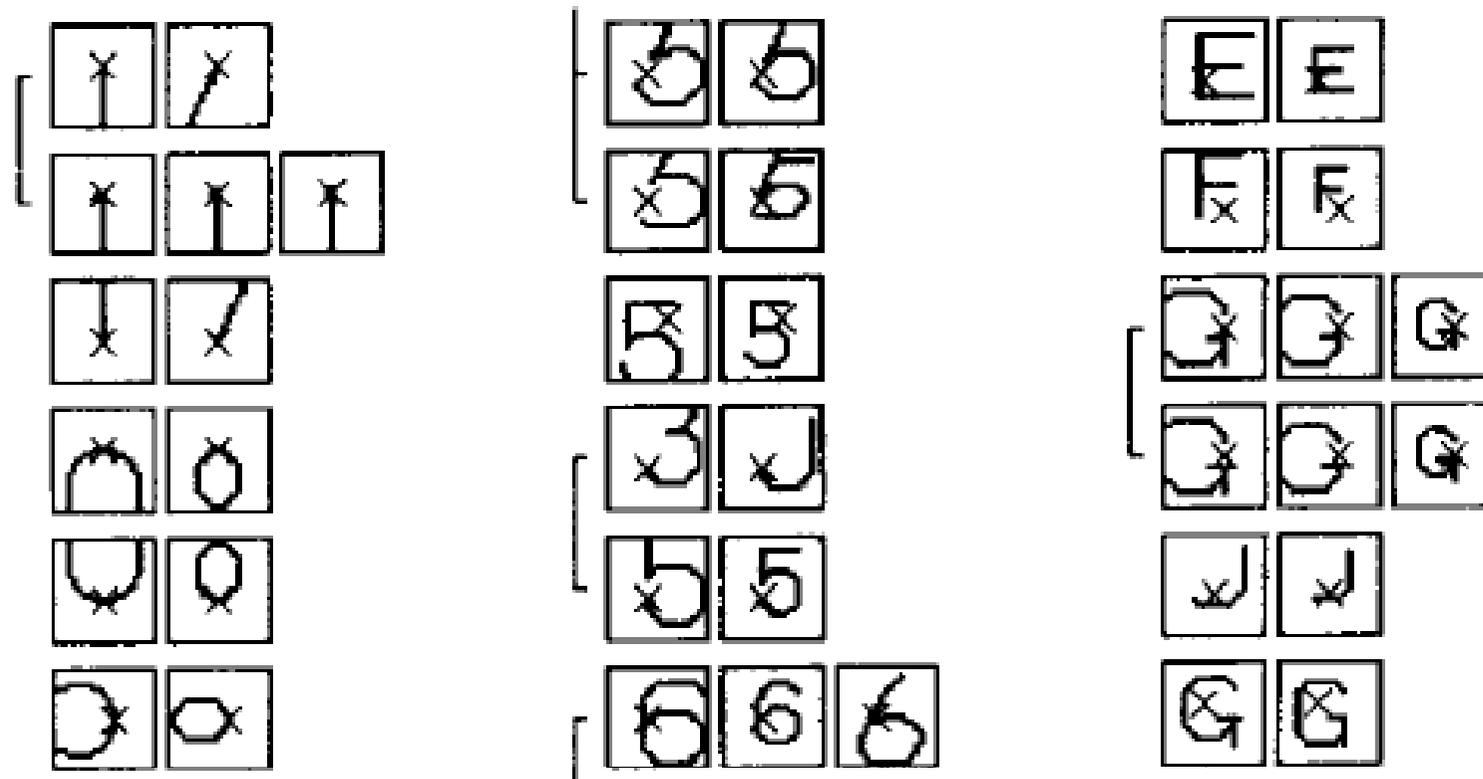
Merkmalsextraktion

- ▶ Beispiel: von U_{S_2} gelernte Muster
 - ▶ Zusammensetzung primitiver Linien der vorherigen Stufe



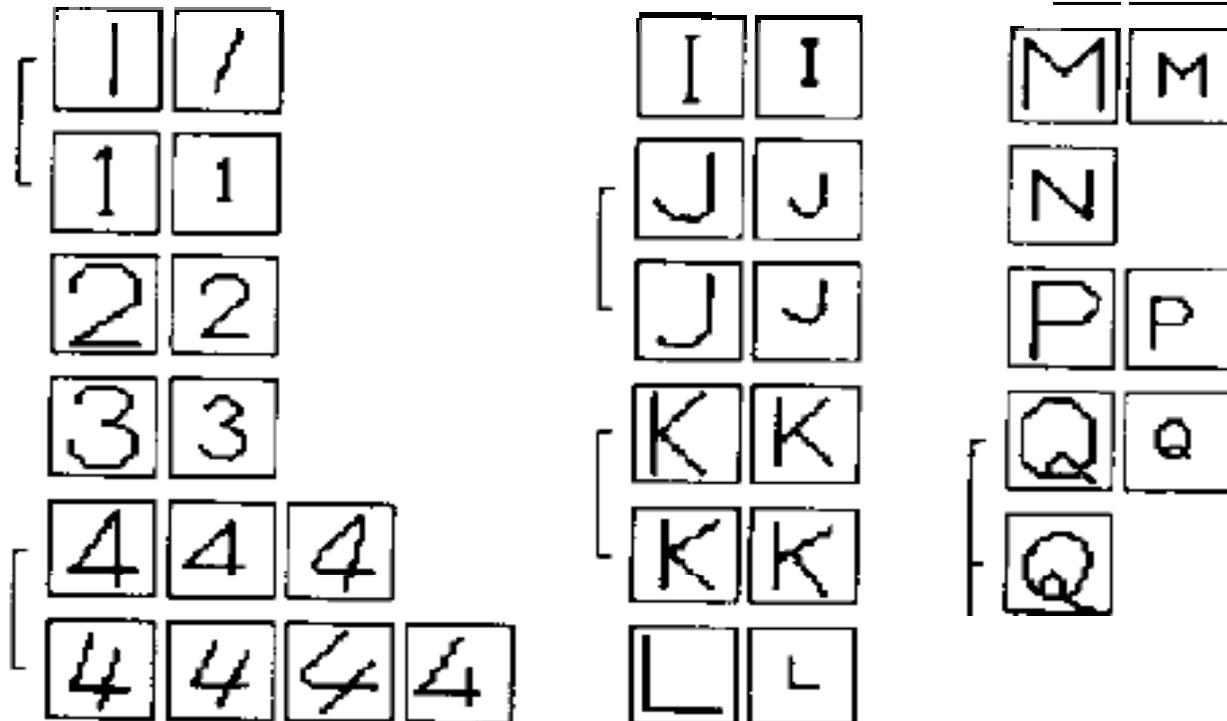
Merkmalsextraktion

- ▶ Beispiel: von U_{S_3} gelernte Muster
 - ▶ Zusammensetzung zu komplexeren Formen / Teilmustern



Merkmalsextraktion

- ▶ Beispiel: von U_{S_4} gelernte Muster
 - ▶ Komplette Muster werden erkannt



(5) Verbindungen und Gewichte



Verbindungen

- ▶ Keine vollständige Verbindung aufeinander folgender Schichten untereinander
- ▶ Verbindungen sind selektiv
- ▶ Verbindungsstruktur hängt allgemein von der Schicht-Art (S- bzw. C-Schicht) ab



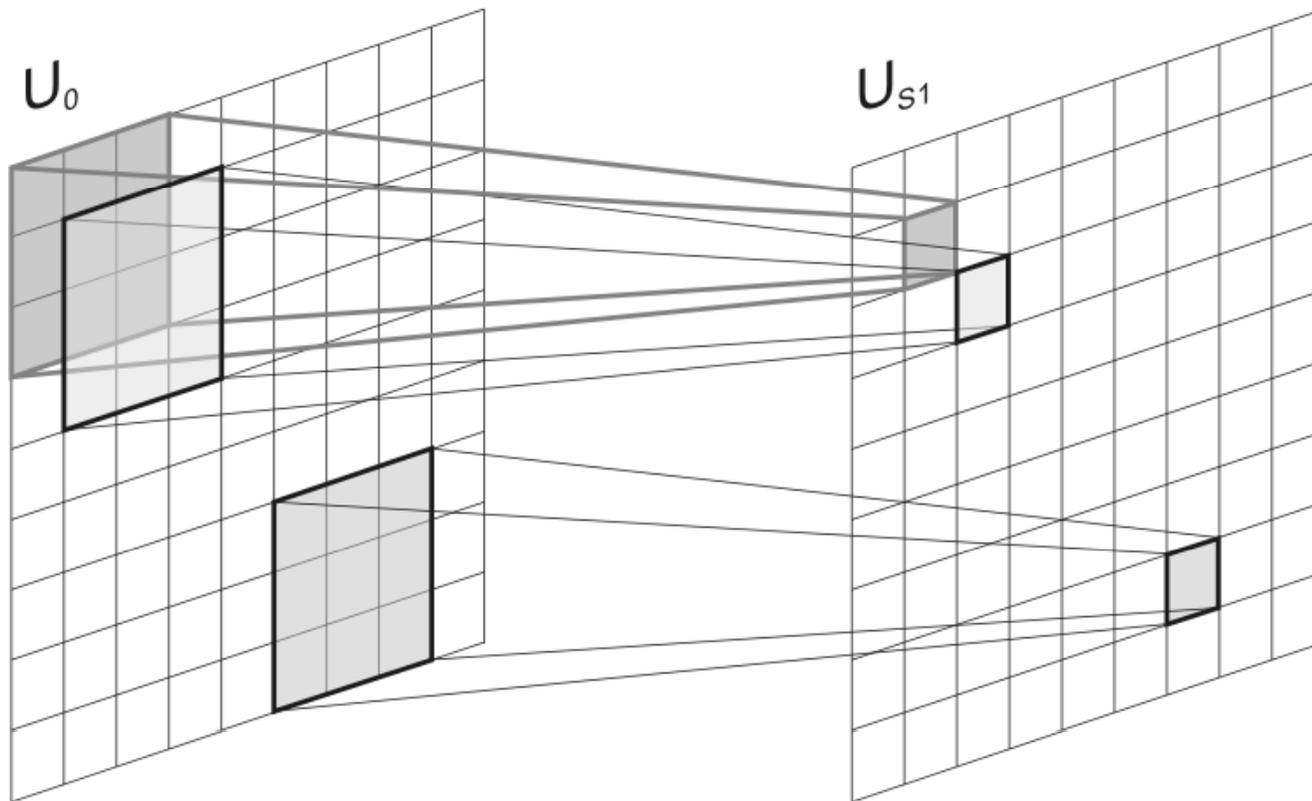
Verbindungen zu S-Schichten

- ▶ Verbindung der einzelnen Ebenen einer S-Schicht mit *allen* C-Ebenen der Vorgängerschicht in gleicher Weise
- ▶ Aufgabe einer S-Ebene: Integration von Einzel-Erkennungen der vorherigen Schicht zu höheren Mustern
- ▶ Verbindung einer S-Zelle mit festgelegtem Bereich an C-Zellen der Vorgängerschicht
- ▶ Benachbarte S-Zellen sind (überlappend) mit benachbarten Bereichen verbunden
- ▶ Alle S-Zellen einer Ebene decken Eingaberaum ab
- ▶ Gewichte sind variabel



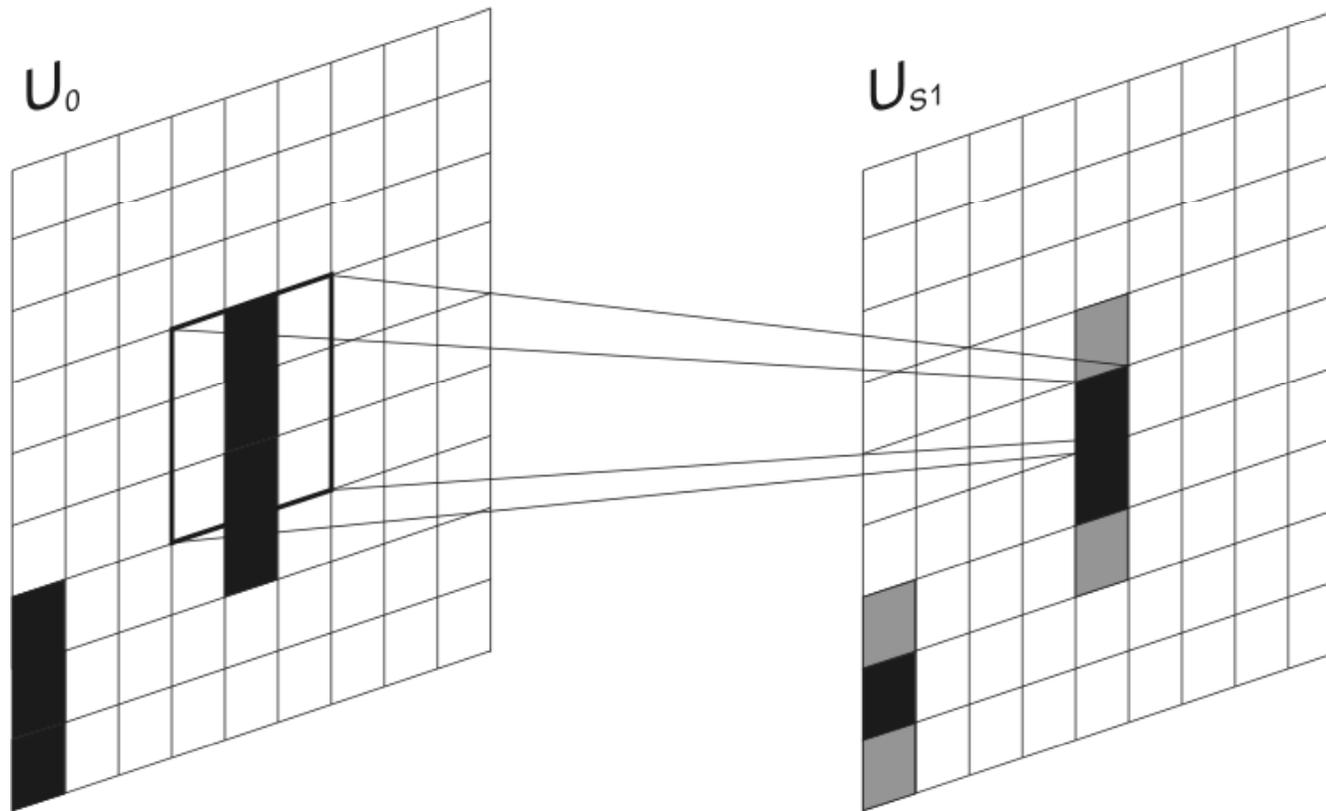
Verbindungen zu S-Schichten

- ▶ Zweck: Verbindung ermöglicht, dass Zellen einer S-Ebene auf dasselbe Muster an *beliebiger* Position im Eingaberaum reagieren



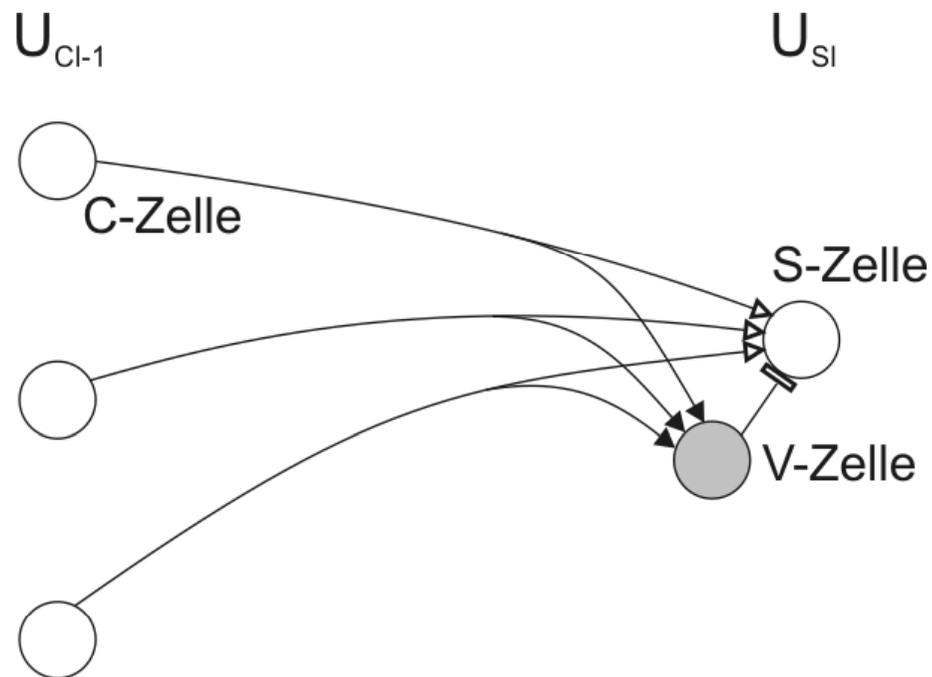
Verbindungen zu S-Schichten

- ▶ Tritt das gelernte Muster an beliebiger Position in der Eingabe auf, reagiert mind. eine S-Zelle der Ebene darauf



Verbindungen zu S-Schichten

- ▶ Wichtiges Detail: V-Zellen
 - ▶ Jeder S-Zelle ist eine V-Zelle zugeordnet
 - ▶ Wirken inhibitorisch auf S-Zellen ein
 - ▶ „Wächter“-Funktion vor irrelevanten Strukturen



Mathematische Beschreibung

- ▶ Ausgabe von V-Zellen:

$$v = \sqrt{\sum_i x_i^2 c_i}$$

- ▶ Ausgabe von S-Zellen:

$$e = \sum_i x_i a_i \quad h = v \cdot b$$

$$y = r \cdot \phi \left(\frac{1 + e}{1 + \frac{r}{1+r} \cdot h} - 1 \right) \quad \text{mit } \phi(z) = \begin{cases} z, & \text{falls } z \geq 0 \\ 0, & \text{sonst} \end{cases}$$



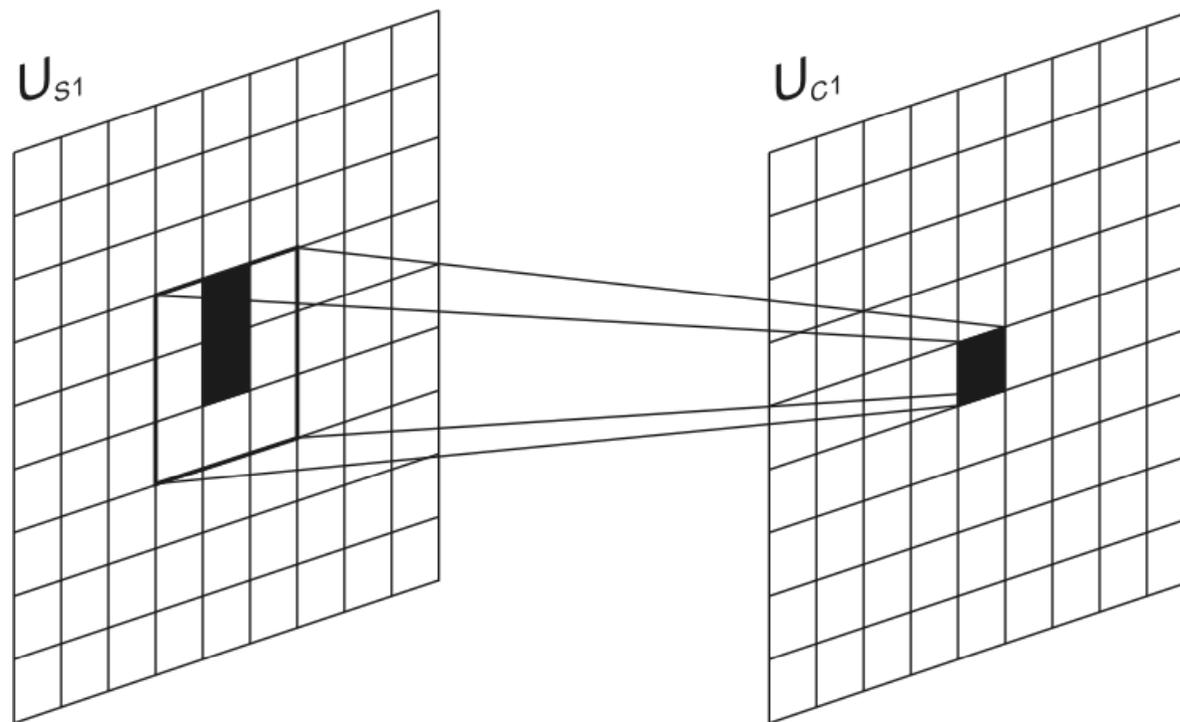
Verbindungen zu C-Schichten

- ▶ Eine C-Ebene ist i.d.R. mit *einer* S-Ebene der Vorgängerschicht verbunden
- ▶ Aufgabe von C-Ebenen: Muster vorhergehender S-Ebenen tolerant machen gegenüber Skalierungen, Verschiebungen und Deformationen
- ▶ Eine C-Zelle erhält Eingaben aus einem kleinen Gebiet
- ▶ Abdeckung des Eingaberaums durch alle C-Zellen einer Ebene (ähnlich S-Zellen)
- ▶ Gewichte sind fest



Verbindungen zu C-Schichten

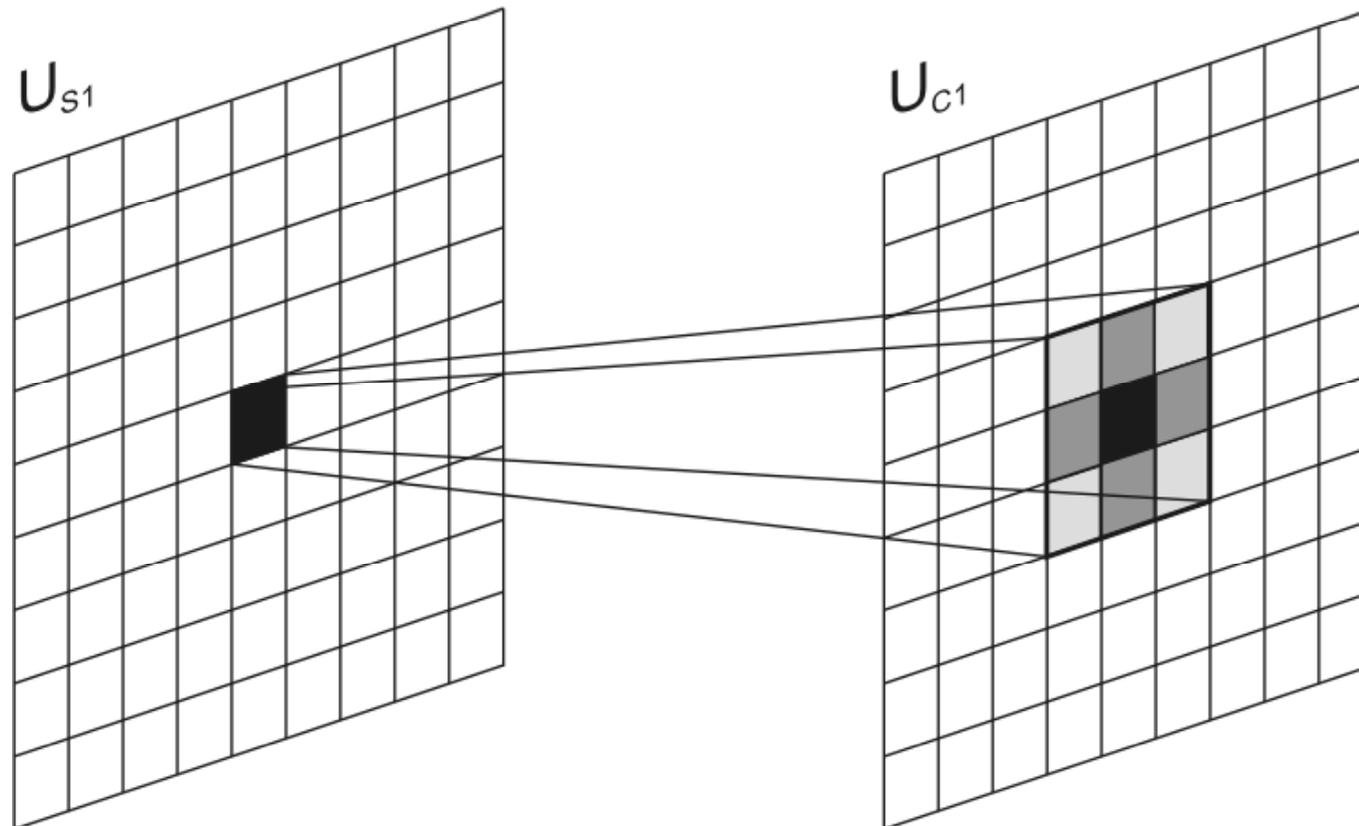
- ▶ Sichtweise 1: eine C-Zelle erhält Eingaben mehrerer S-Zellen



- ▶ Zweck: Muster wird auch noch durch C-Zelle erkannt, wenn es leicht verschoben ist → Toleranz
 - ▶ positionsbedingter Fehler
-

Verbindungen zu C-Schichten

- ▶ Sichtweise 2: eine S-Zelle aktiviert mehrere C-Zellen

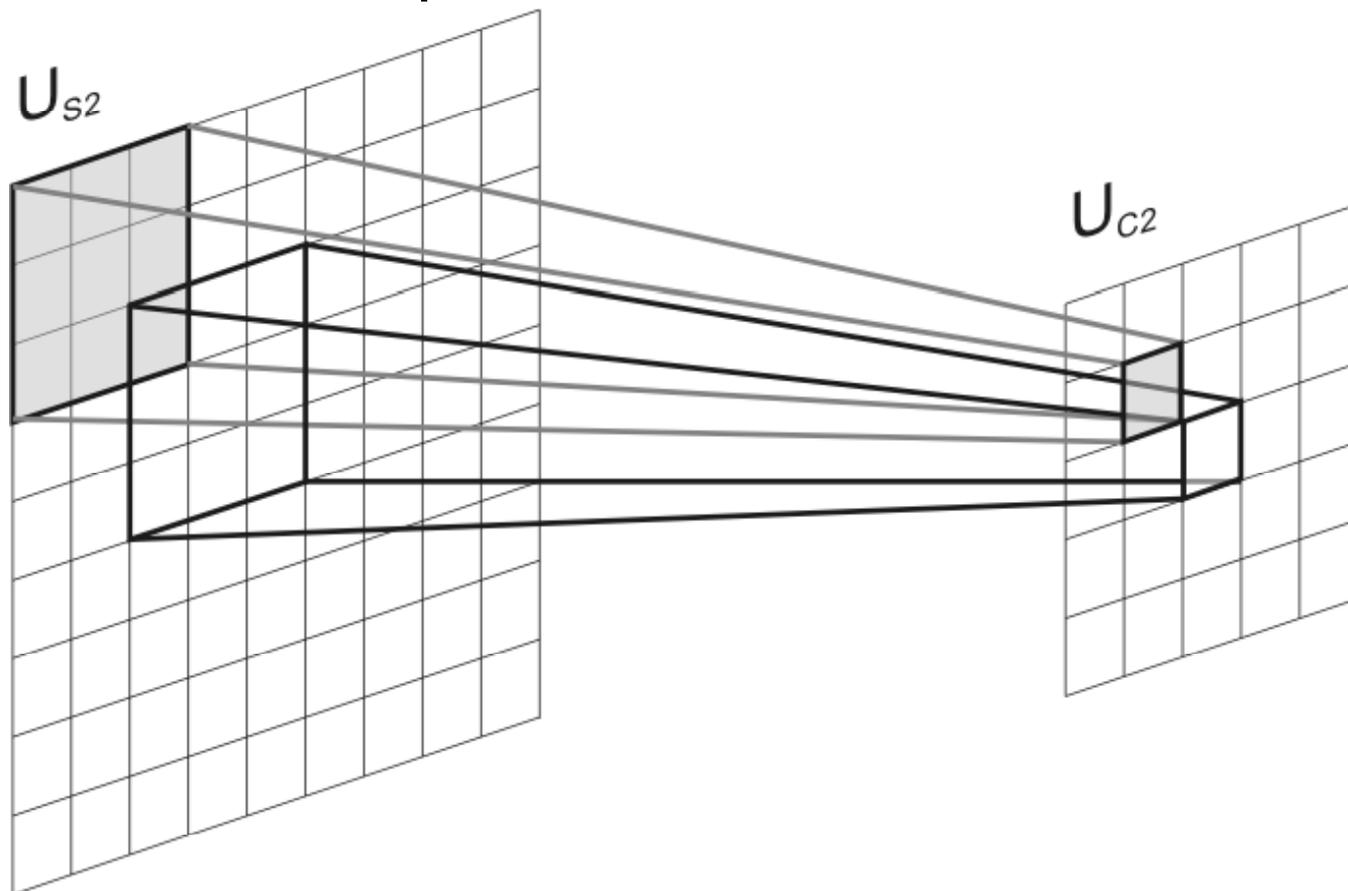


- ▶ Zweck: Glättung der Erkennung → fügt Toleranz gegenüber Skalierungen und Deformationen hinzu
-



Verbindungen zu C-Schichten

- ▶ Weitere wichtige Eigenschaft: Verkleinerung der Zellenanzahl pro Ebene durch C-Schichten



Mathematische Beschreibung

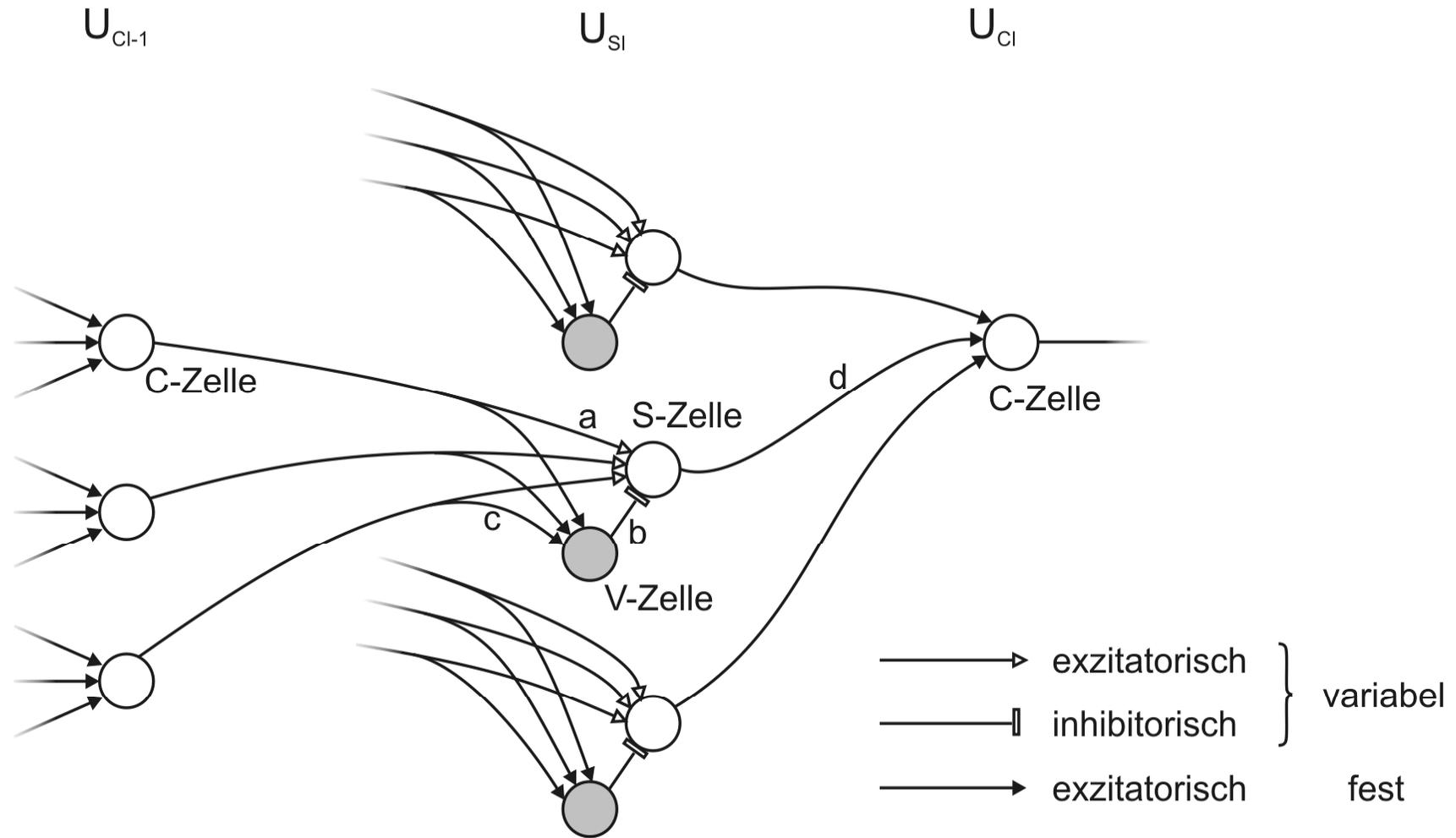
- ▶ Ausgabe von C-Zellen:

$$x = \psi \left(\sum_j y_j d_j \right)$$

$$\text{mit } \psi(z) = \frac{\phi(z)}{1 + \phi(z)} \text{ und } \phi(z) = \begin{cases} z, & \text{falls } z \geq 0 \\ 0, & \text{sonst} \end{cases}$$



Verbindungen im Überblick

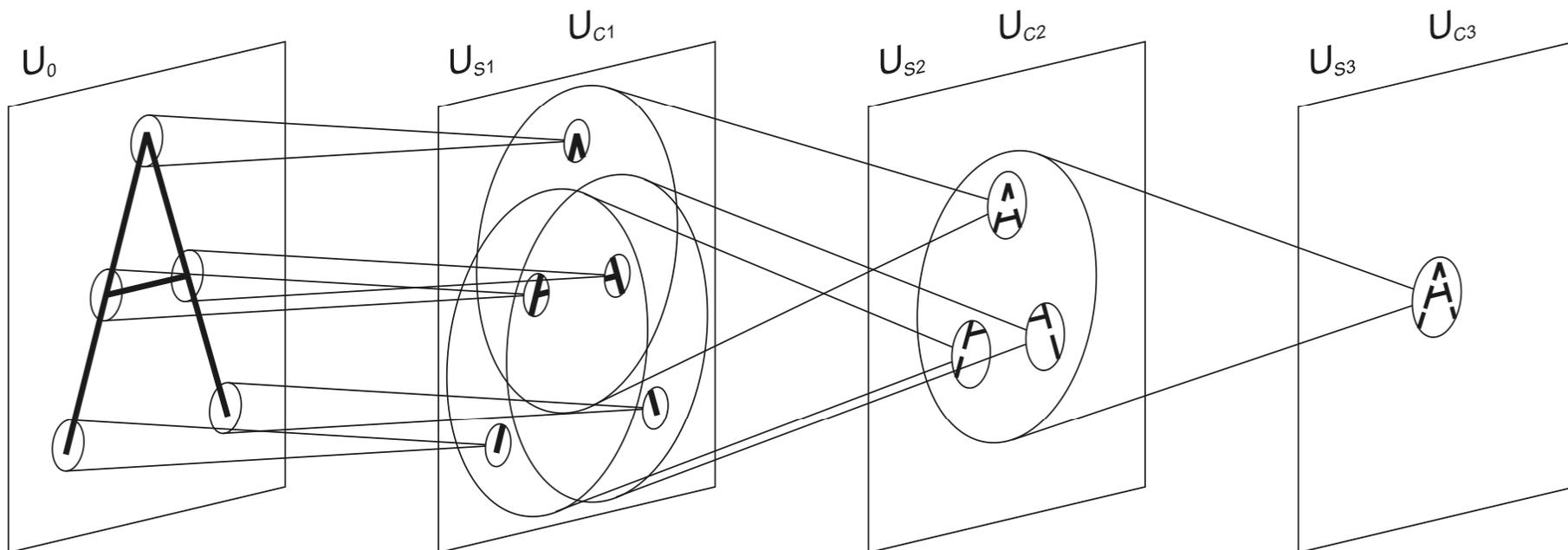


(6) Prozess der Mustererkennung



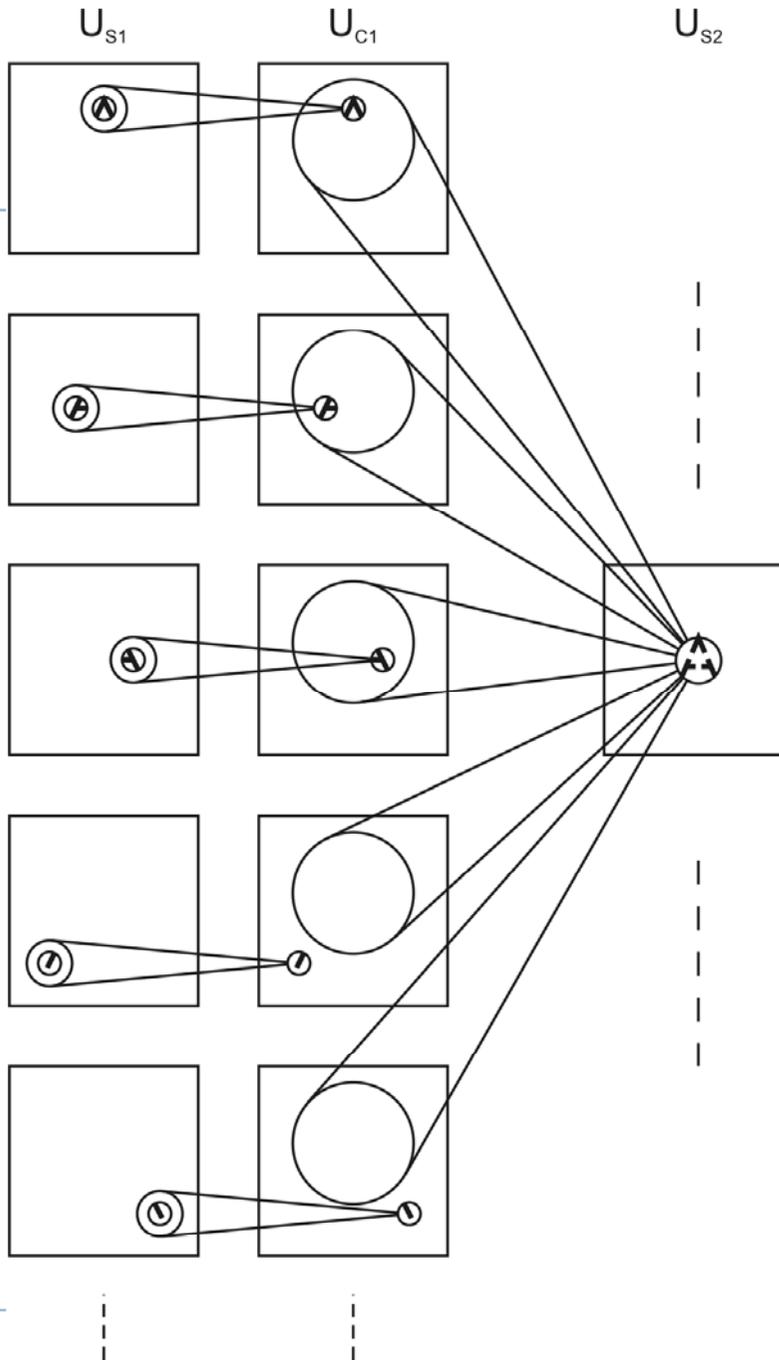
Prozess der Mustererkennung

- ▶ Untere Schichten erkennen: einfache Strukturen
- ▶ Höhere Schichten: komplexere Strukturen
- ▶ Beispiel: Buchstabe A



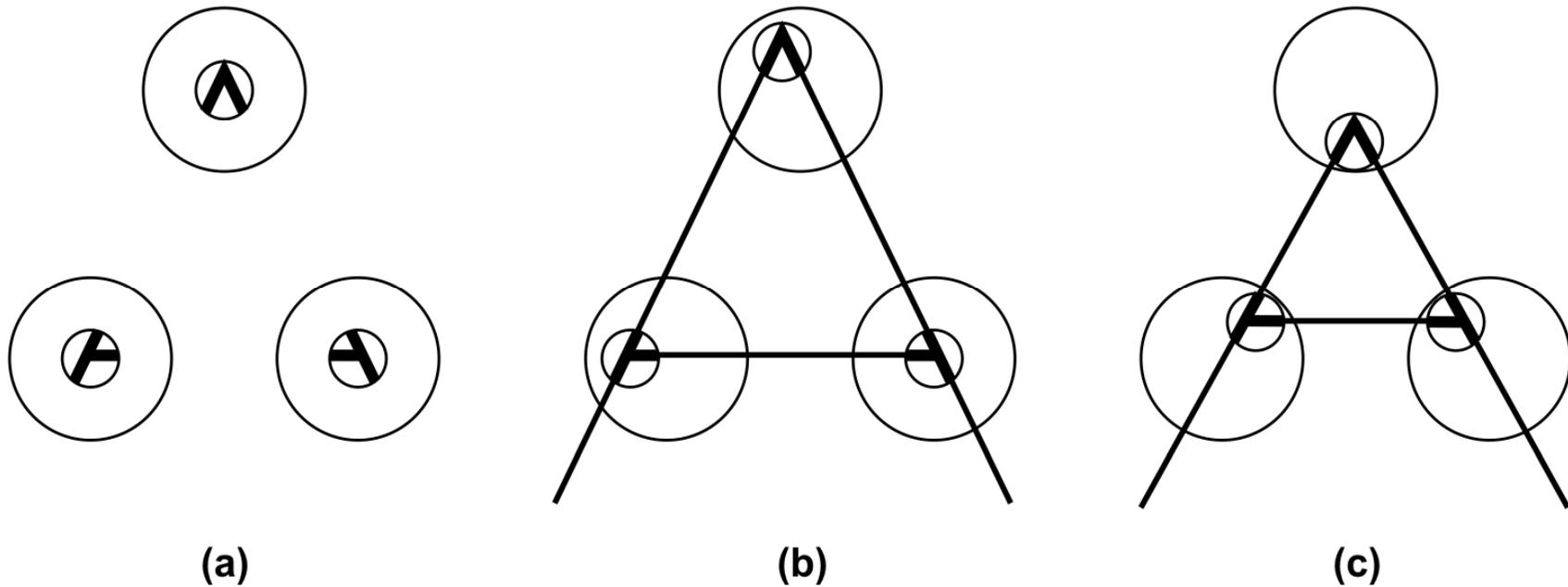
(6)

- ▶ Genauere Betrachtung der 1. Stufe:

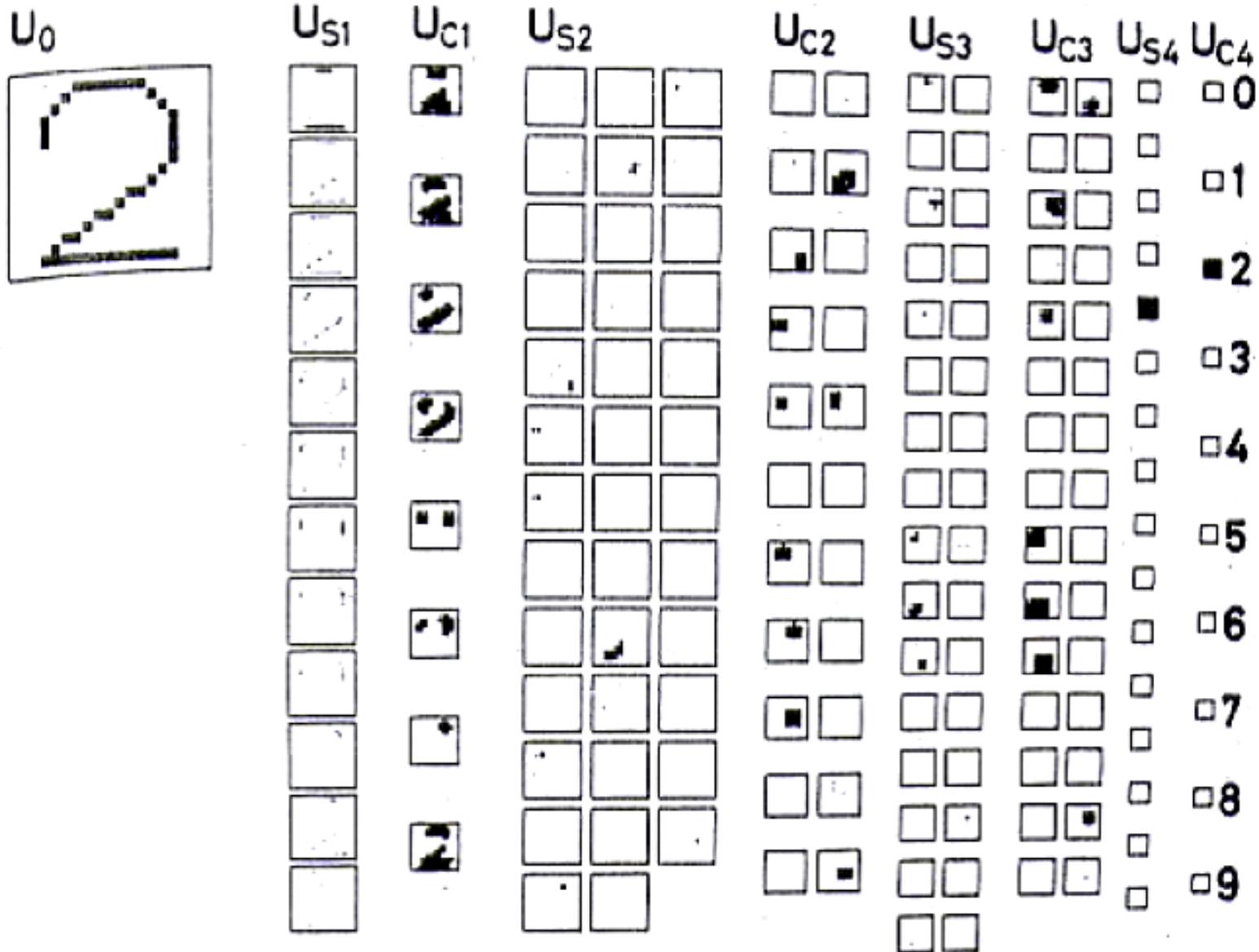


Prozess der Mustererkennung

- ▶ Hinzufügen von Toleranzen durch C-Schichten:



Erkennungs-Beispiel



Erkennungs-Beispiel 2

U₀



U_{c1} U_{c2} U_{c3} U_{c4}

								<input type="checkbox"/>	0
								<input type="checkbox"/>	1
								<input checked="" type="checkbox"/>	2
								<input type="checkbox"/>	3
								<input type="checkbox"/>	4
								<input type="checkbox"/>	5
								<input type="checkbox"/>	6
								<input type="checkbox"/>	7
								<input type="checkbox"/>	8
								<input type="checkbox"/>	9

(7) Überwachtes Lernen



Überwachtes Lernen

- ▶ Schichtenweises Lernen
 - ▶ „Lernen mit Lehrer“:
 - ▶ Bei vorwärtsgerichtetem Netz:
 - ▶ Lehrer präsentiert Eingabe- und Ausgabedaten, Netz passt sich diesen Daten an
 - ▶ Beim Neocognitron:
 - ▶ Lehrer geht die Ebenen einer Schicht durch, wählt für jede Ebene getrennt das zu lernende Teilmuster
 - ▶ „vollständig überwachtes Training“
 - ▶ Großer Aufwand in der Erstellung sinnvoller Trainingsdaten
 - ▶ **Bessere Klassifizierung als bei unüberwachtem Lernen**
-

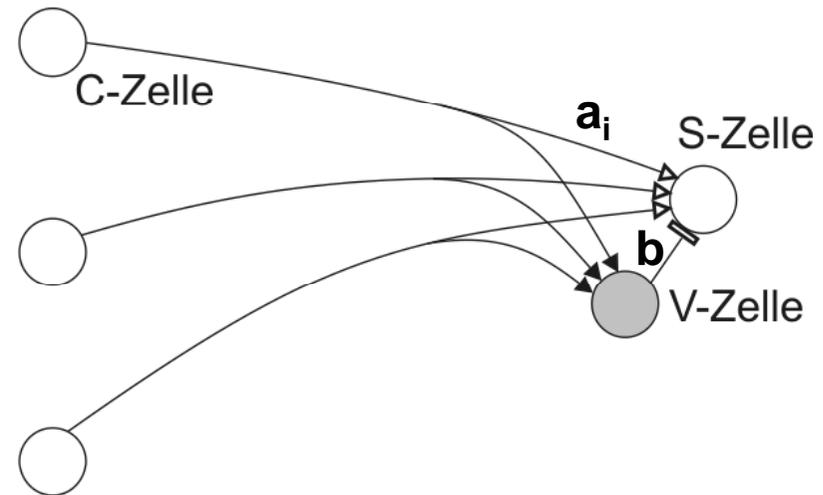


Überwachtes Lernen

- ▶ In einer Ebene: Lehrer wählt Keimzelle, deren Gewichte angepasst werden mit:

$$\Delta a_i = q \cdot a_i \cdot x_i$$

$$\Delta b = q \cdot v$$



- ▶ Andere Zellen der Ebene erhalten dieselben Gewichte (*weight sharing*)
→ reagieren somit auf dasselbe Muster
-

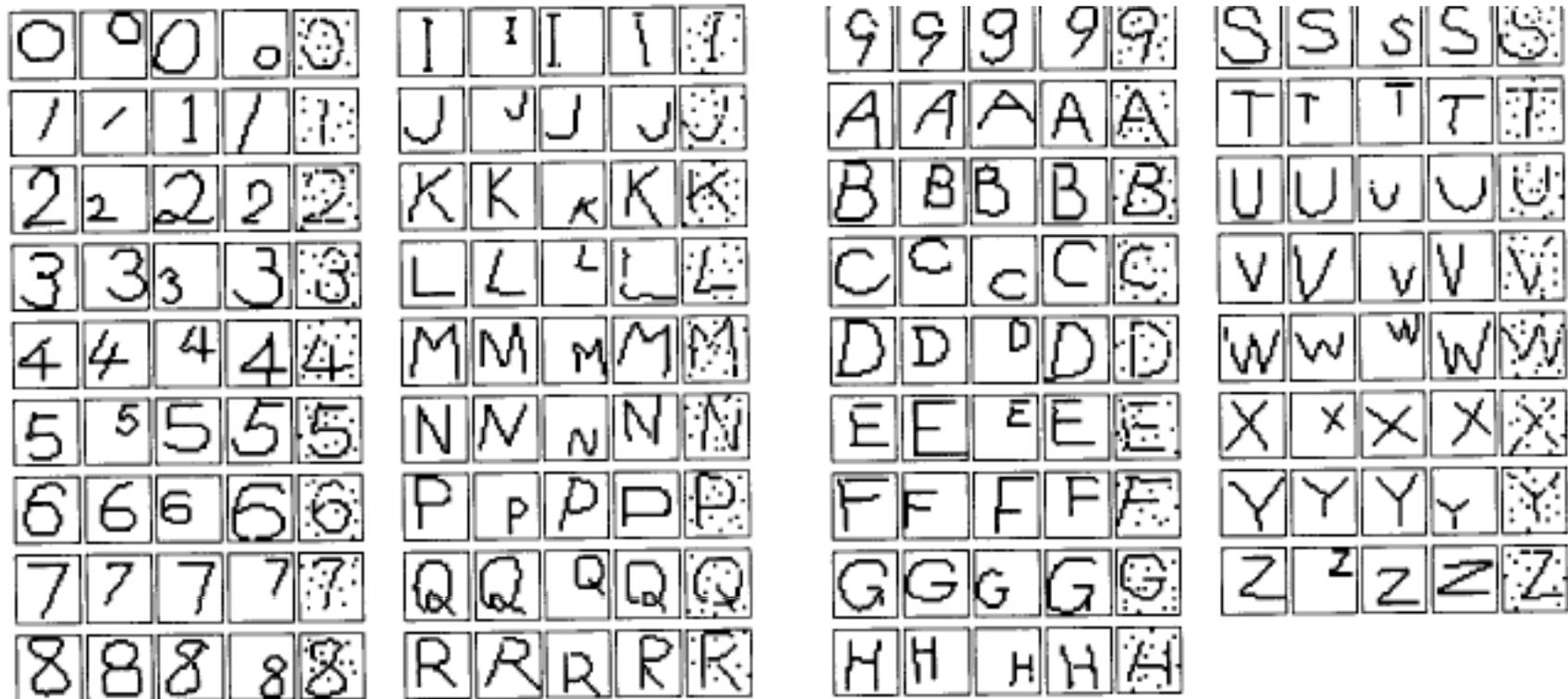


(8) Modifikationen und Anwendungsgebiete



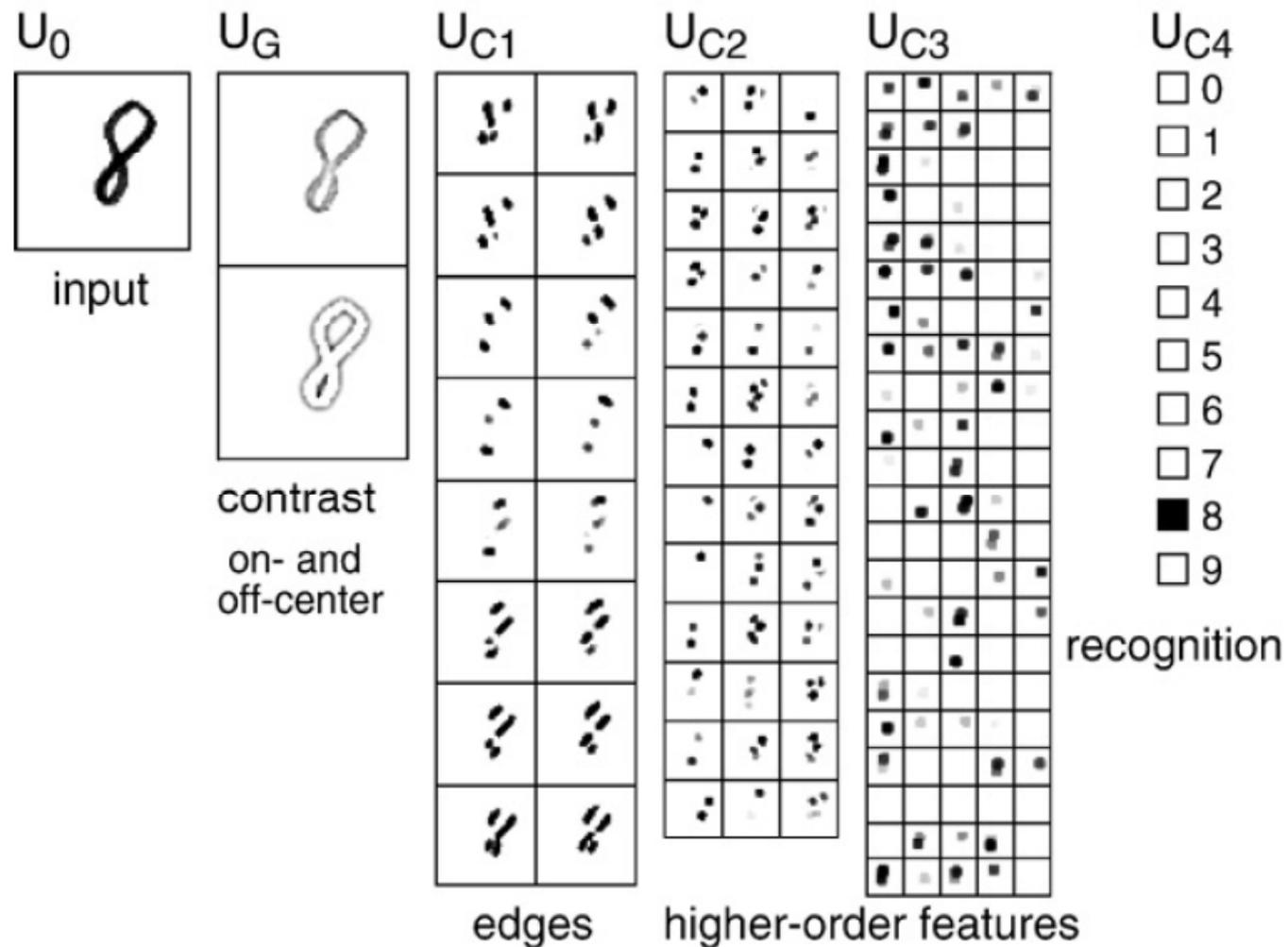
Modifikationen/Anwendungen

- ▶ Erkennung handschriftlicher Zeichen (Fuku91):



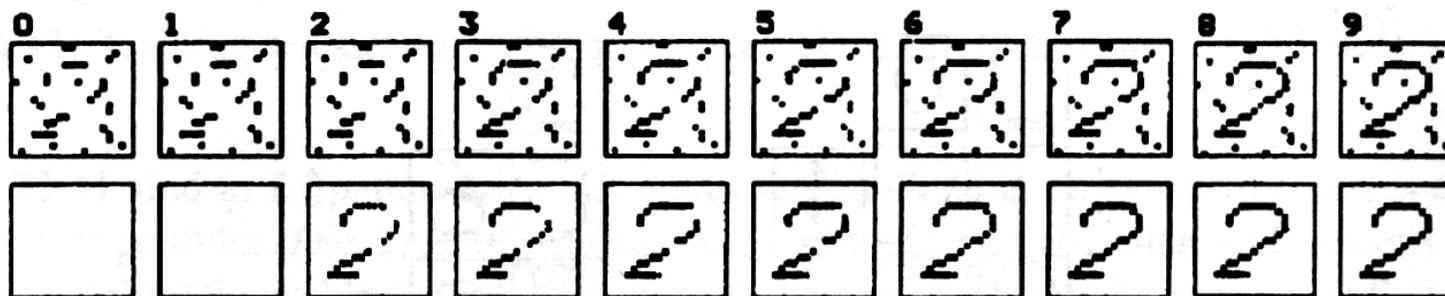
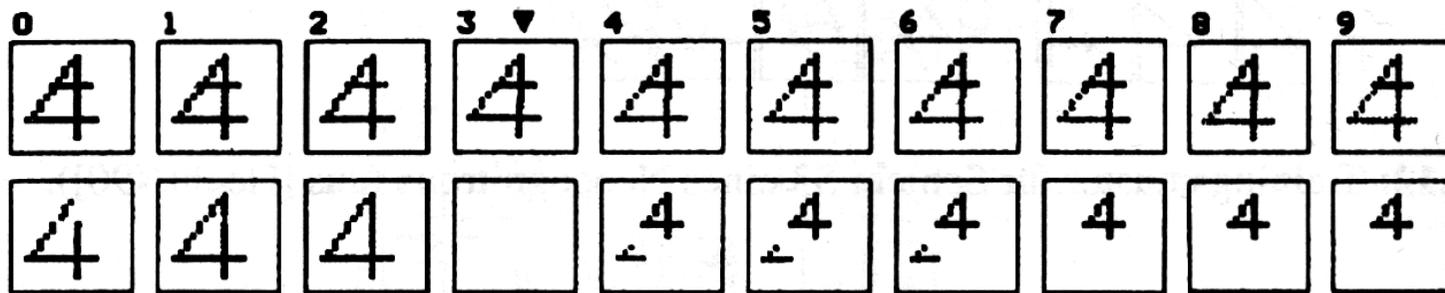
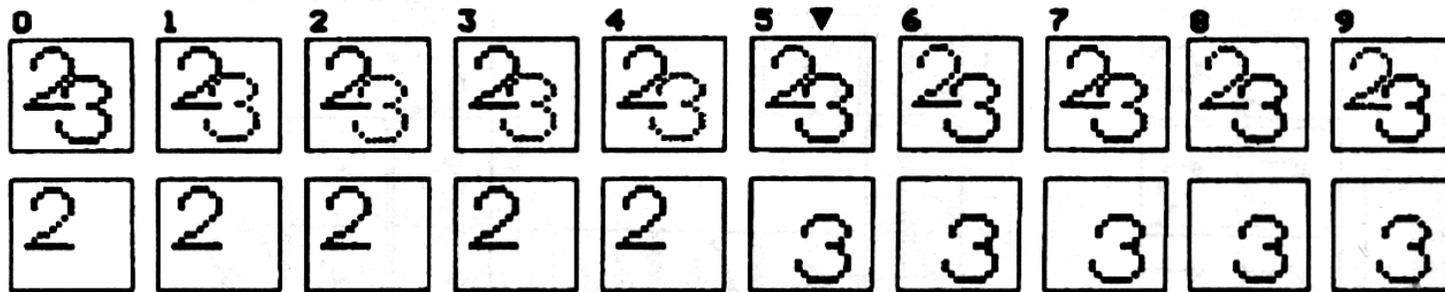
Modifikationen/Anwendungen

- ▶ Zahlreiche Verbesserungen, z.B. Fuku03:



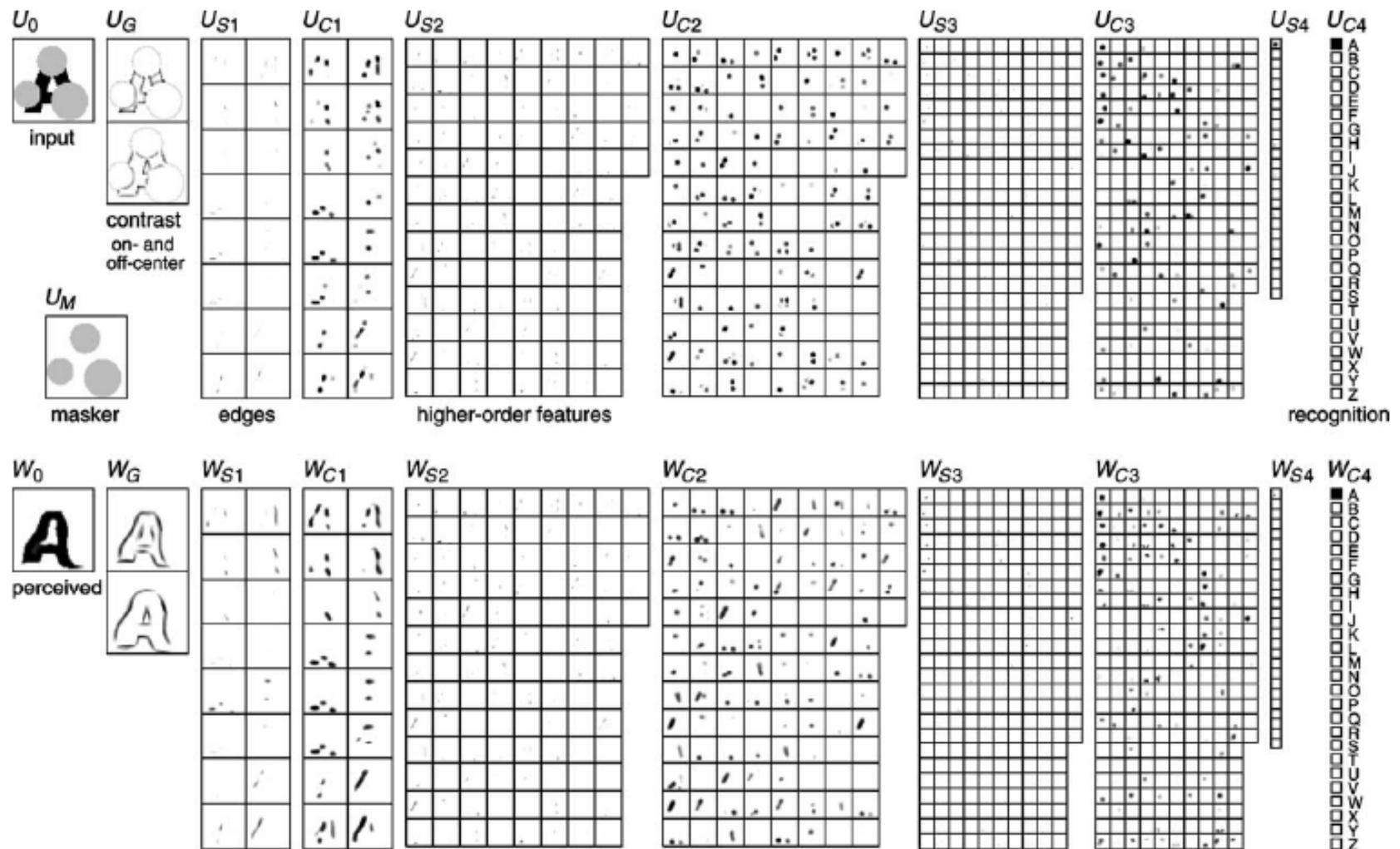
Modifikationen/Anwendungen

▶ Selective Attention Mechanismus:



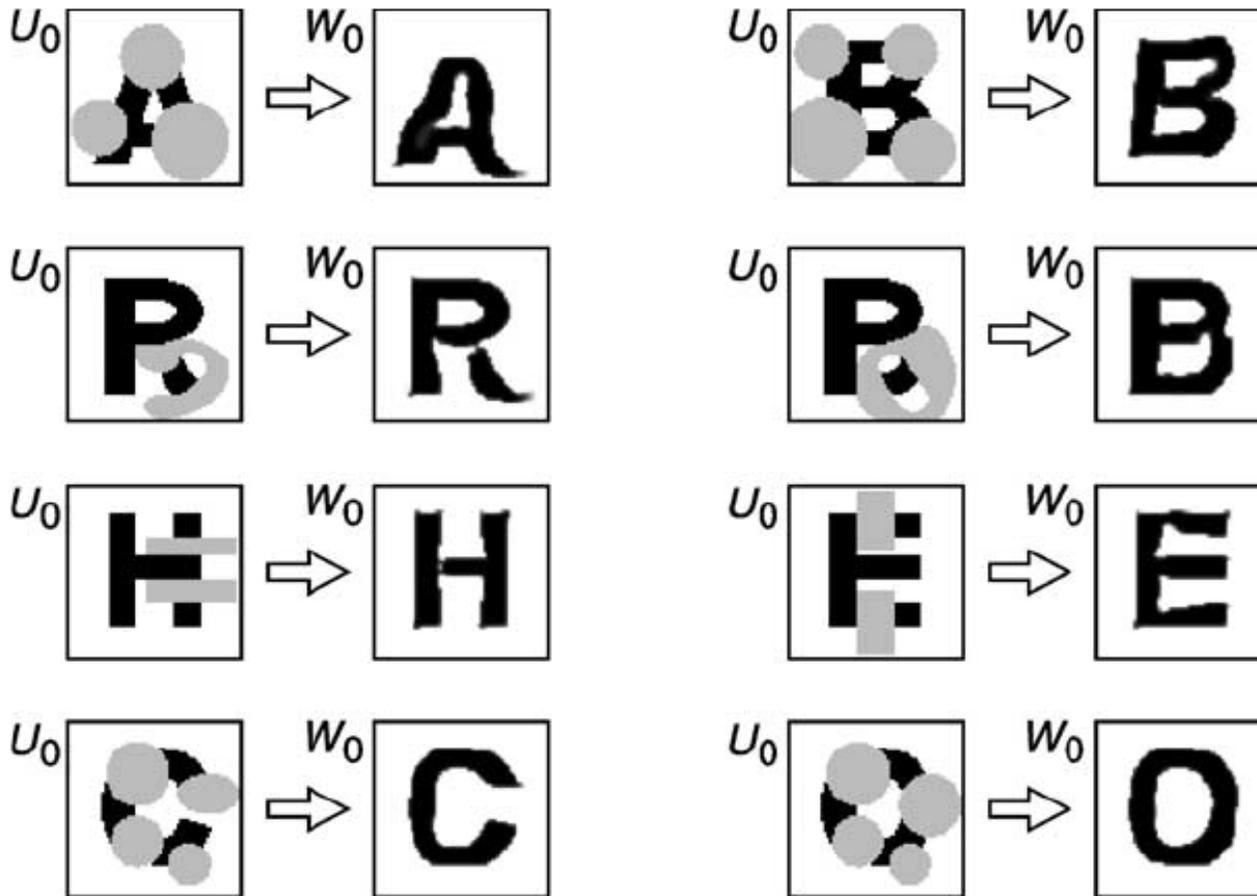
Modifikationen/Anwendungen

▶ Verdeckte Muster (Fuku05):



Modifikationen / Anwendungen

► Verdeckte Muster (Fuku05):



(9) Zusammenfassung



Zusammenfassung

- ▶ Neocognitron: hierarchisches mehrschichtiges KNN zur Mustererkennung
- ▶ Modell des visuellen Wahrnehmungssystems
- ▶ Tolerierung von Skalierungen, Deformationen und Verschiebungen

- ▶ Keine (kaum) Tolerierung von Rotationen
- ▶ Aufwändige Netz-Modellierung (Parameter etc.) und Erstellung eines Trainings-Datensatzes



Ausblick

- ▶ Gegenstand weiterer Verbesserungen
- ▶ Weiterhin fruchtbare Wechselwirkung mit Kognitionsforschung
- ▶ Ziel 1: Modell des Wahrnehmungssystems erstellen
- ▶ Ziel 2: Computersimulation erzeugen, die Fähigkeiten des Menschen nahekومت



Ende

