

EBERHARD KARLS  
UNIVERSITÄT  
TÜBINGEN



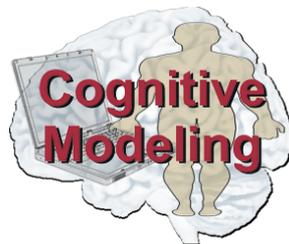
TECHNICAL REPORT  
No. COBOSLAB\_Y2014\_N001  
January 9, 2014

---

ANSÄTZE ZUR MODELLIERUNG VON SIMULTANAGNOSIE

---

JOHANNES M. KURZ, ESTHER F. KUTTER  
BETREUER:  
HANS-OTTO KARNATH, ANNA BELARDINELLI, MARTIN V. BUTZ



WILHELM-SCHICKARD-INSTITUT FÜR INFORMATIK  
UNIVERSITY OF TÜBINGEN  
SAND 14  
72076 TÜBINGEN, GERMANY  
[HTTP://WWW.CM.INF.UNI-TUEBINGEN.DE](http://www.cm.inf.uni-tuebingen.de)



# Ansätze zur Modellierung von Simultanagnosie

Johannes M. Kurz\*      Esther F. Kutter†

**Betreuer:**

Hans-Otto Karnath‡      Anna Belardinelli§      Martin V. Butz¶

## 1 Motivation

Die Mechanismen, welche der *Simultanagnosie*, einer neurologischen Störung, bei der Patienten den Wald vor lauter Bäumen nicht sehen“ können, zugrunde liegen, sind nach wie vor nicht vollständig erforscht. Ein möglicher Erklärungsansatz wäre, dass durch die Hirnläsionen wichtige ”Schaltstellen“, die den Wechsel zwischen globalem und lokalem Aufmerksamkeitsfokus regulieren, beeinträchtigt sind.

Basierend auf quantitativen Theorien (Riesenhuber & Poggio, 1999; Serre et al., 2005) zum ventralen visuellen Verarbeitungspfad haben Serre, Wolf, Bileschi, Riesenhuber, and Poggio (2007) das biologisch motivierte HMAX-Modell zur Objekterkennung entwickelt. Mit einer Erweiterung dieses Modells um die oben vorgeschlagenen aufmerksamkeitsregulierenden Mechanismen sollten sich die Befunde der Simultanagnosie-Studien simulieren lassen.

Die vorliegende Arbeit soll dabei als Vorstudie dienen, in der grundsätzliche Möglichkeiten und Grenzen des HMAX-Modells untersucht werden, die im Hinblick auf spätere Erweiterungen von Bedeutung sind.

### 1.1 Neuropsychologischer Hintergrund

Als geübter Kartenspieler, der er war, hätte er beim ersten Blick erkennen müssen, was für eine Kartenkombination er in der Hand hatte. Er versagte aber dabei, obwohl er jede einzelne Karte richtig erkannte.

(Wolpert, 1924, S.405f.)

---

\*johannes.kurz@student.uni-tuebingen.de

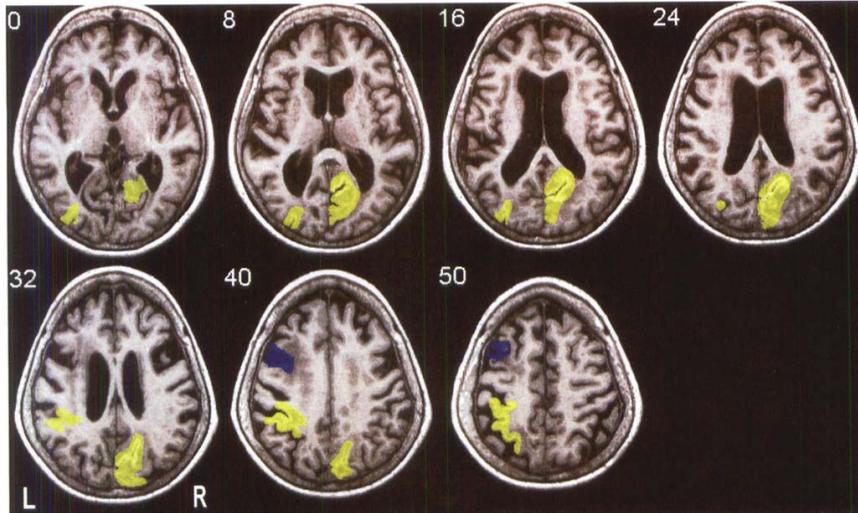
†esther.kutter@student.uni-tuebingen.de

‡karnath@uni-tuebingen.de

§belardinelli@informatik.uni-tuebingen.de

¶martin.butz@uni-tuebingen.de

Diese Unfähigkeit, mehrere Objekte gleichzeitig und als Bestandteil einer Gesamtszene wahrzunehmen – obwohl sie einzeln sehr gut erkannt werden – wird als *Simultanagnosie* bezeichnet (Wolpert, 1924). Sie tritt häufig im Kontext des Blint-Holmes Syndroms (Bálint, 1909) in Kombination mit Blickbewegungsstörungen, Störungen der räumlichen Orientierung und optischer Ataxie auf.



**Figure 1:** Simultanagnosie tritt häufig infolge von bilateralen Läsionen des parieto-okzipitalen Kortex (im Bild gelb hervorgehoben) auf (Karnath et al., 2000).

Die Störung tritt zumeist infolge von bilateralen Läsionen des posterior parietalen Kortex (PPC), einer wichtigen Region des dorsalen visuellen Verarbeitungspfad, auf (siehe Abbildung 1). Die genauen zugrunde liegenden Prozesse sind aber noch ungeklärt.

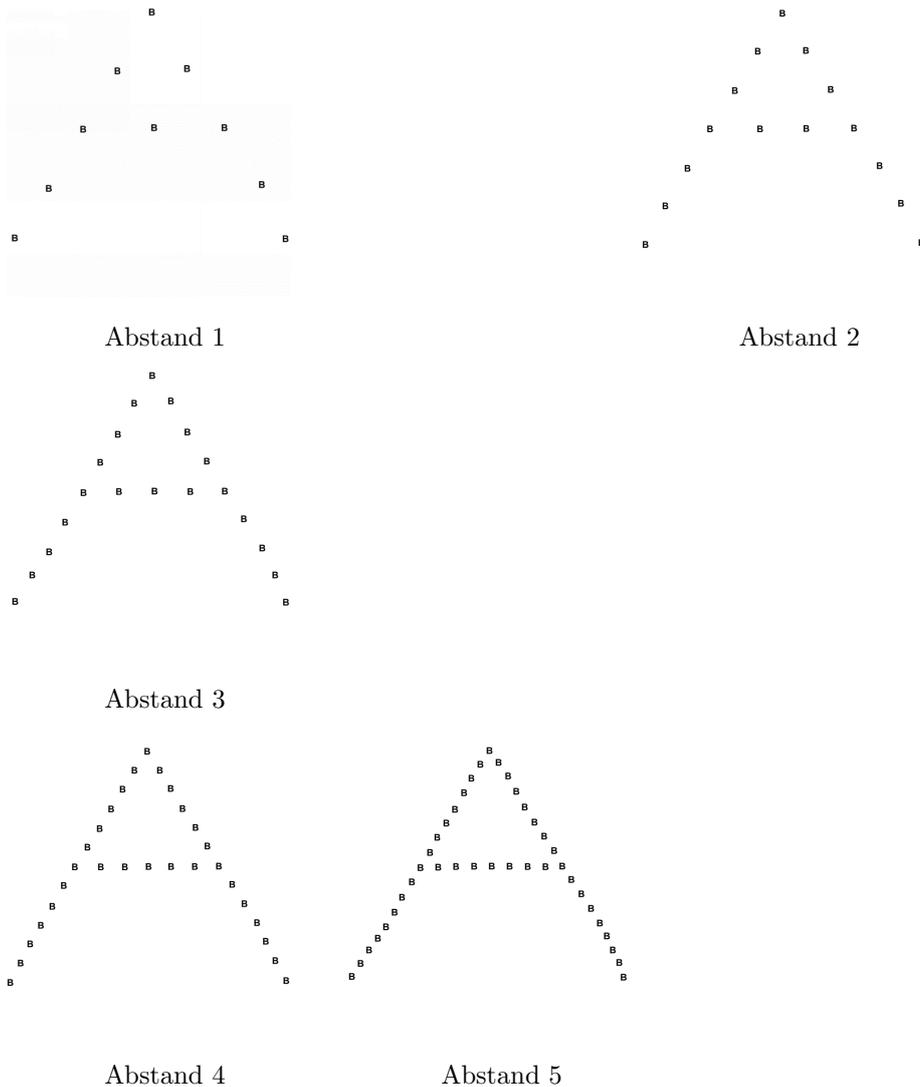
Um diese Mechanismen gezielt untersuchen zu können, werden den Patienten daher häufig hierarchisch strukturierte, so genannte Navon-Buchstaben präsentiert, bei denen ein großer (globaler) Buchstabe aus mehreren kleinen (lokalen) Buchstaben zusammengesetzt wird (Navon, 1977) (siehe Abbildung 2). Gesunde Menschen sind problemlos in der Lage, die lokalen Elemente zu einem ganzheitlichen Bild zusammenzusetzen und das globale Element zu benennen. Die Patienten hingegen können auch hierbei wieder nur die einzelnen kleinen Buchstaben identifizieren, selbst wenn man sie bittet, sich nur auf das globale Objekt zu konzentrieren.

Interessanterweise scheint die Integration multipler Objekte nicht per se unmöglich zu sein. So erkannte ein Patient zwar immer nur eines von zwei aufgemalten Objekten (das andere ignorierte er vollständig), verband man die Bilder aber durch eine Linie, konnte er plötzlich das gesamte Bild erkennen (Luria, 1959). Dies legt den Schluss nahe, dass die räumlichen Relationen zwischen den lokalen Objekten einen entscheidenden Einfluss

## 1 MOTIVATION

---

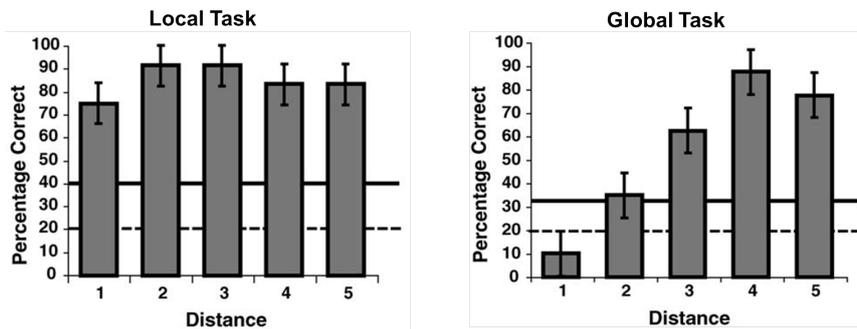
auf die globale Erkennungsleistungen haben könnten. Um diese Hypothesen zu untersuchen, präsentierten Huberle und Karnath (2006) den Patienten Navon-Buchstaben, in denen der Abstand zwischen den einzelnen kleinen Buchstaben variierte. Die Größe der lokalen und des globalen Buchstaben wurde dabei konstant gehalten (siehe Abbildung 2).



**Figure 2:** Um den Einfluss des Abstandes zwischen lokalen Objekten auf die globale Objekterkennung zu untersuchen, verwendeten Huberle and Karnath (2006) Navon-Buchstaben mit konstanten Buchstabengrößen, aber variabler Dichte der lokalen Elemente.

Die Patienten wurden nun gebeten, entweder den großen (Global Task) oder die kleinen (Local Task) Buchstaben zu identifizieren. Erwartungs-

gemäßgelang es ihnen fast immer, die kleinen Buchstaben korrekt zu benennen. In der Global Task zeigte sich nun ein erstaunliches Antwortmuster: Waren die Buchstaben dicht beieinander, so konnte auch der globale Buchstabe deutlich über Chance Level identifiziert werden (siehe Abbildung 3). Diese Resultate belegen, dass die globale Objekterkennung kein Alles-oder-Nichts-Phänomen ist, sondern maßgeblich durch die räumliche Dichte lokaler Elemente beeinflusst werden kann.



**Figure 3:** Ergebnisse der Studien von Huberle and Karnath (2006): Kleine Buchstaben werden unabhängig von ihrem Abstand zueinander stets sehr gut erkannt (links). Erstaunlicherweise kann mit höherer Buchstabendichte auch der große Buchstabe zunehmend besser identifiziert werden (rechts). Gesunde Kontrollpersonen erreichen in beiden Aufgaben Erkennungsleistungen um 100%.

Daneben gibt es noch weitere Studien, die sich mit dem Verhältnis zwischen globalen und lokalen Elementen befassen beispielsweise Huberle, Driver, and Karnath (2010); Huberle and Karnath (2012). Da in dieser Arbeit aber das Stimulusmaterial von Huberle and Karnath (2006) verwendet wurde und auch erste Simulationen mit diesen Ergebnissen überstimmten, wurde diese Arbeit an dieser Stelle besonders erwähnt.

Globale und lokale Objekterkennung wird also offensichtlich durch verschiedene neuronale Strukturen umgesetzt, wobei die Verschaltung beider Prozesse bzw. der bewusste Wechsel zwischen dem jeweiligen Aufmerksamkeitsfokus durch Areale reguliert wird, die bei Simultanagnosie-Patienten beeinträchtigt sind. Durch nachfolgende Studien, in denen verschiedenartig modifizierte Navon-Buchstaben verwendet wurden, rücken einige Areale, beispielsweise die Temporoparietal Junction (TPJ), der Anteriore Cinguläre Kortex (ACC), der Precuneus oder auch der Lateral Geniculate Nucleus (LGN) im Thalamus, die das Zusammenspiel von ventralem und dorsalem visuellen Pfad mit regulieren könnten, besonders in den Fokus (Huberle & Karnath, 2012; Thomas, Kveraga, Huberle, Karnath, & Bar, 2012).

Modellsimulationen könnten diesbezüglich weiter Aufschluss geben.

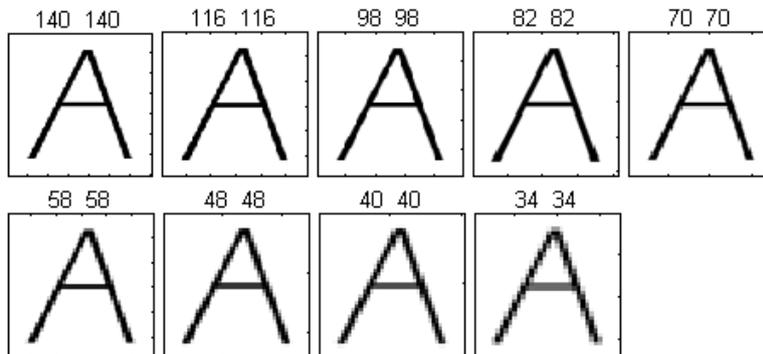
## 1.2 Modelltheoretischer Hintergrund

Ein sehr bekanntes Modell zur Objekterkennung stellt die HMAX-Architektur dar (Riesenhuber & Poggio, 1999; Mutch & Lowe, 2006; Serre et al., 2007), die analog zur strukturellen Organisation des visuellen Kortex und ventralen Pfades aufgebaut ist. Sie beruht auf den folgenden, allgemein anerkannten Annahmen zur Verarbeitung visueller Informationen (siehe hierzu auch Serre et al., 2005):

- Die Verarbeitung erfolgt *hierarchisch* – von einfacher Kantendetektion bis hin zu komplexeren Stimuli–, wodurch eine gewisse Positions- und Größen*invarianz* erreicht werden kann.
- Die ersten Verarbeitungsstufen, zur unmittelbaren Erkennung ohne Aufmerksamkeitsverschiebungen, sind ausschließlich *feedforward*.
- *Lernprozesse* finden auf allen Hierarchiestufen statt, insbesondere aber auf der höchsten Ebene im inferotemporalen (IT) und präfrontalen Kortex (PFC).

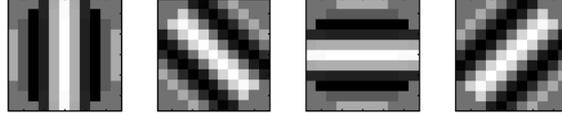
Das HMAX-Modell besteht dabei aus einer Eingabeschicht, gefolgt von sich abwechselnden Neuronenschichten, die in Anlehnung an die von Hubel and Wiesel (1959) entdeckten kortikalen Zellen als Simple- (S) und Complex-Layers (C) bezeichnet werden:

**Image Layer.** Das Eingabebild wird in ein Graustufenbild der Größe 140x140 px umgewandelt. Davon wird anschließend eine 9-stufige Bilderpyramide erstellt, wobei die nächste Stufe jeweils um den Faktor  $2^{1/4}$  kleiner ist (siehe Abbildung 4).



**Figure 4:** Bilderpyramide, die exemplarisch für den in Abbildung 12 links dargestellten Buchstaben erstellt wird.

**Gabor Filter (S1) Layer.** In Analogie zu den rezeptiven Feldern der Simple Cells im V1 werden verschiedene Gaborfilter (siehe Abbildung 5) an allen möglichen Positionen der gesamten Bilderpyramide platziert.



**Figure 5:** Das Modell arbeitet mit vier je 11x11 px großen Gaborfiltern, die 90°, 135°, 0° und 45° (von links nach rechts) orientiert sind.

Für jede Lokation werden dann die Aktivierungen der einzelnen Gaborfilter gespeichert (siehe Abbildung 6). Zu beachten ist, dass nur die Skalierung der Bilder geändert wird; die Filter sind stets 11x11 px groß. Die Aktivierung eines Bildpatches  $X$  bei einem S1-Filter  $G$  wird folgendermaßen bestimmt:

$$R(X, G) = \left| \frac{\sum X_i G_i}{\sqrt{\sum X_i^2}} \right| \quad (1)$$

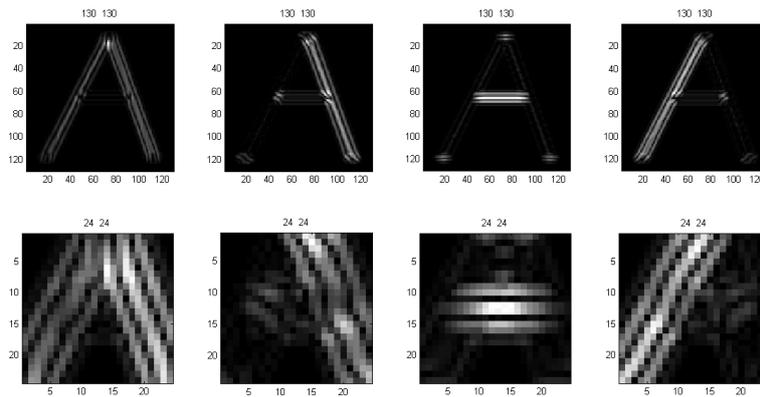
wobei

$$G(x, y) = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} X\right) \quad (2)$$

mit

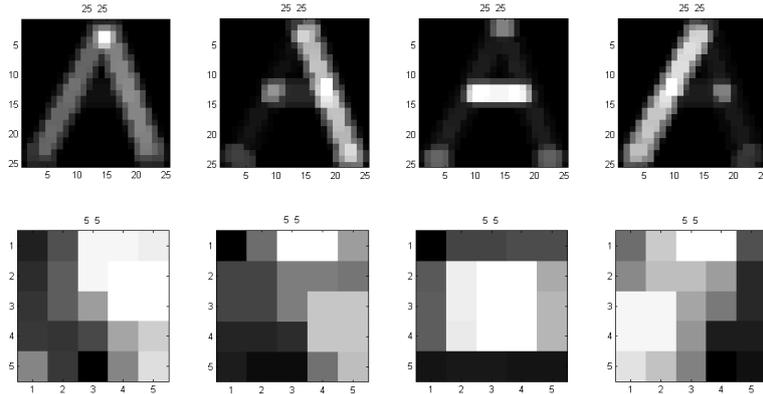
$$X = x \cos \theta - y \sin \theta \quad \text{und} \quad Y = x \sin \theta + y \cos \theta \quad (3)$$

Die Parameter  $\gamma = 0.3$  (Aspect Ratio),  $\sigma = 4.5$  (Effective Width) und  $\lambda = 5.6$  (Wavelength) sowie  $x, y \in [-5, +5]$  und  $\theta \in [0, \pi]$  wurden aus Mutch and Lowe (2006) übernommen.



**Figure 6:** Aktivierungen des S1-Layers für die größte (oben) und kleinste (unten) Skalierung, jeweils beim Anlegen der in Abbildung 5 gezeigten Gaborfilter.

**Local Invariance (C1) Layer.** Auf dieser Ebene wird lokale Positions- und Größeninvarianz erreicht. Dabei wird für jede Lokation jeweils der maximale Wert zweier benachbarter S1-Stufen (gleicher Orientierung) innerhalb des Max-Filters gespeichert (siehe Abbildung 7). Dadurch werden jeweils zwei Stufen zusammengefasst und somit die benötigte Stufen-Anzahl im C1-Layer verringert. Diese Ebene modelliert die Complex Cells im V1.



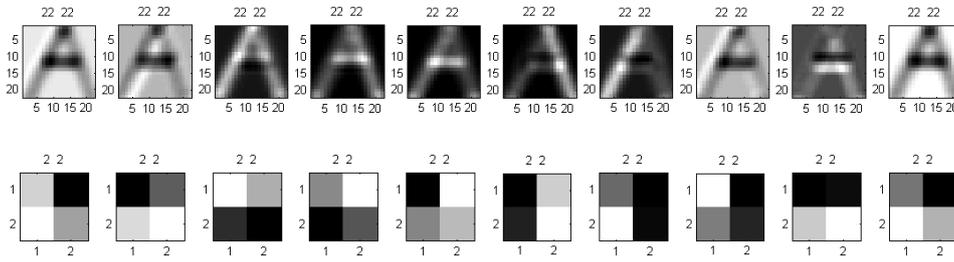
**Figure 7:** Aktivierungen des C1-Layers für die größte (oben) und kleinste (unten) Skalierung nach dem Anlegen des Max-Filters. Die Werte werden für jede Orientierung separat gespeichert.

**Intermediate Feature (S2) Layer.** Diese Ebene entspricht etwa dem kortikalen V4-Areal bzw. dem inferotemporalen Kortex. Hier werden die im C1-Layer gespeicherten Werte für sämtliche Positionen und Skalen mit vorher ermittelten Prototypen (siehe Abbildung 9) mittels Radialer Basisfunktionen verglichen (die Lernphase wird im Folgenden näher beschrieben). Die Aktivierung (siehe Abbildung 8) eines C1-Patches  $X$  bei einem S2-Prototypen  $P$  der Dimension  $N \times N \times 4$  (mit  $N \in \{4, 8, 12, 16\}$ ) wird folgendermaßen bestimmt:

$$R(X, P) = \exp\left(-\frac{\|X - P\|^2}{2\sigma^2\alpha}\right) \quad (4)$$

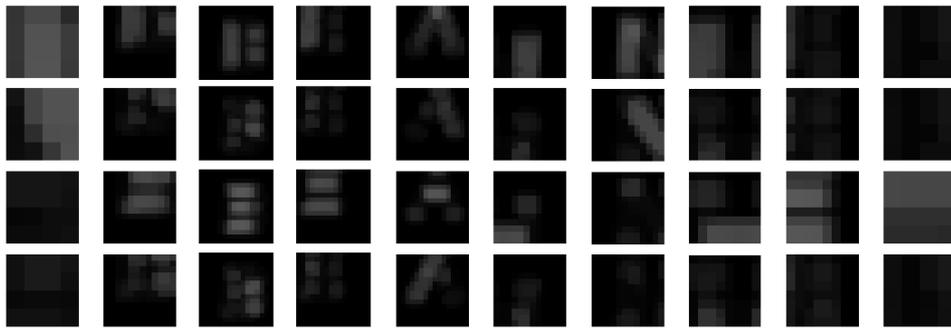
Die Standardabweichung  $\sigma$  wurde auf 1 gesetzt;  $\alpha = (N/4)^2$  dient als Normalisierungsfaktor.

**Global Invariance (C2) Layer.** Auf dieser Ebene wird ein Vektor generiert, der für jeden Prototypen dessen maximale Antwort an einer beliebigen Bildposition und Skalengröße speichert, wodurch auch globale Positions- und Größeninvarianz erreicht wird.



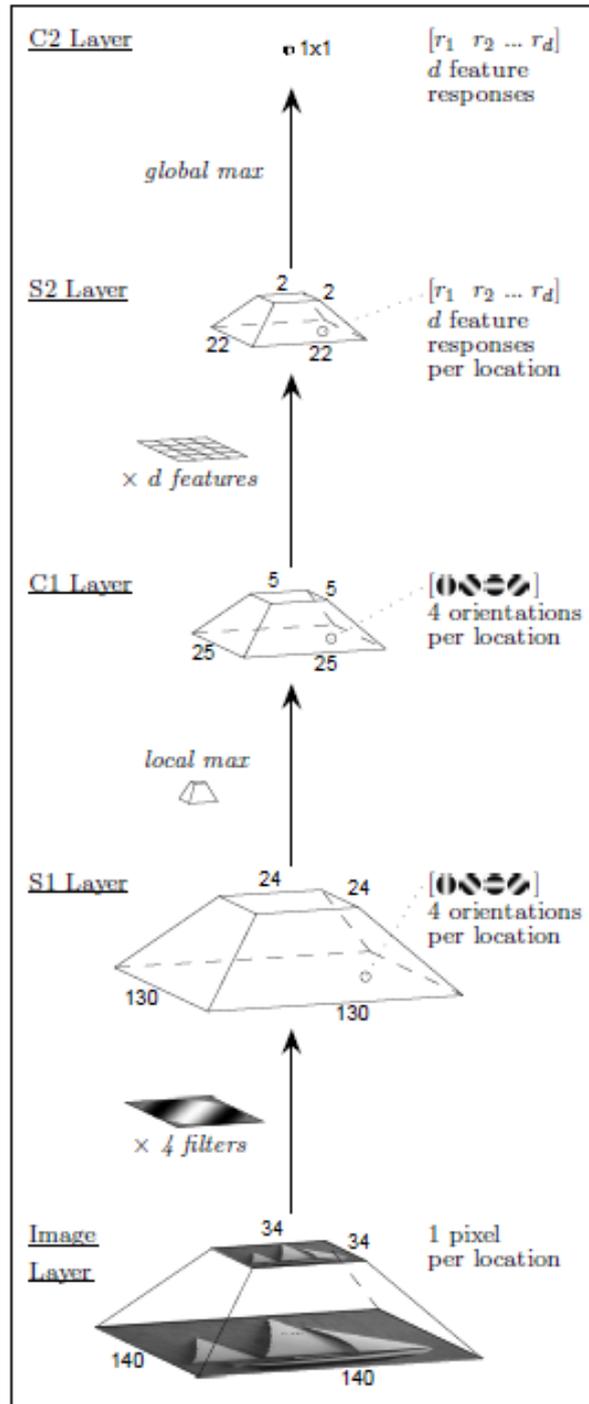
**Figure 8:** Aktivierung des S2-Layers für die größte (oben) und kleinste (unten) Skalierung beim Vergleich mit den Prototypen (siehe auch Abbildung 9). Exemplarisch sind die Werte für 10 zufällig ausgewählte Prototypen dargestellt.

Im S2-Layer werden die aus dem Bild extrahierten Features mit zuvor abgespeicherten Prototypen verglichen, die in einer anfänglichen **Lernphase** erstellt und in einem globalen Feature Dictionary gespeichert werden. Dazu werden Bildpatches verschiedener Größen (4x4, 8x8, 12x12, 16x16px) an zufälligen Positionen und Skalen mehrerer Eingabebilder platziert, die Werte des entsprechenden C1-Layers ausgelesen und als Prototypen gespeichert. In Abbildung 9 sind einige Prototypen exemplarisch dargestellt.



**Figure 9:** Zehn zufällig ausgewählte Prototypen (Spalten) unterschiedlicher Größen. In einem Prototyp der Größe  $N$  ist die Aktivierung an  $N \times N$  Lokationen für jede der vier Orientierungen (Zeilen) gespeichert.

Die C2-Vektoren werden nun durch eine lineare One-vs-Rest Support Vektor Maschine (SVM) **klassifiziert**: Nach einer **Trainingsphase**, in welcher der SVM Bilder verschiedener Kategorien gezeigt werden, werden die **Testbilder** durch Mehrheitsvoting einer der gegebenen Klassen zugeordnet.



**Figure 10:** HMAX-Architektur nach Mutch and Lowe (2006). Neben den Skalengrößen der einzelnen Bilderpyramiden sind auch die Dimensionen der auf diesen Ebenen gespeicherten Werte angegeben.

## 2 Modellierung der neuropsychologischen Befunde

### 2.1 Allgemeiner Modellansatz

Die HMAX-Architektur ist bereits ein mächtiges Modell, mit dem sich selbst so komplexe Objekte wie Fahrzeuge, Personen oder Gesichter identifizieren lassen (Serre et al., 2007). Um mit diesem nun auch die Wahrnehmungsdefizite von Simultanagnosie-Patienten simulieren zu können, soll es um verschiedene Komponenten erweitert werden:

(1) Es wird eine Art **Aufmerksamkeitsfilter** integriert, der auf den Annahmen basiert, dass

- nur Reize, die sich innerhalb des Aufmerksamkeitsfokus befinden, auch wahrgenommen und kognitiv verarbeitet werden; alles übrige folglich unbemerkt bleibt, und
- der Fokus bewusst sowohl selektiv auf einzelne Reize als auch global auf die gesamte Szene gelenkt werden kann.

Dieser Filter könnte durch Einbeziehung von bottom-up **Saliency Maps** (bspw. nach Koch & Ullman, 1987; Itti & Koch, 2001) sowie top-down **Erwartungen** hinsichtlich bestimmter Lokationen und Features (Chikkerur, Serre, Tan, & Poggio, 2010) beeinflusst werden.

(2) Aktive **Selbstinhibition** würde die Verarbeitung weiter dynamisieren und die neuronale Sensitivität für neue Stimuli erhöhen. Demnach sollten vor allem folgende Module integriert werden:

- Regulation der Auflösung des Aufmerksamkeitsfokus. Dieser Mechanismus scheint bei Simultanagnosie-Patienten gestört. Neuronal könnte er daher im posterior parietalen Kortex (PPC) verankert sein (siehe hierzu auch Huberle & Karnath, 2012).
- Verschiebung des Fokus weg von der aktuellen Lokation und den aktuellen Features. Der inferotemporale Kortex (IT) scheint dabei eine entscheidende Rolle zu spielen (Chikkerur et al., 2010).

Sollten sich mit diesen Erweiterungen die Wahrnehmungsdefizite erfolgreich simulieren lassen können, so könnte dies zum Verständnis der visuellen Verarbeitungsprozesse und ihrer neuronalen Korrelate beitragen.

### 2.2 Zielstellung der vorliegenden Arbeit

Die vorliegende Arbeit kann im Wesentlichen als Vorstudie betrachtet werden, in der allgemeine Möglichkeiten und Grenzen des Modells im Hinblick auf die Simultanagnosie-Befunde untersucht werden sollen.

Von den oben erläuterten Komponenten soll daher vorerst nur der erste Punkt – die Integration eines Aufmerksamkeitsfilters – in einer sehr vereinfachten Form berücksichtigt werden (siehe hierzu Abschnitt 4). Die oben erwähnten top-down Priors wurden in dieser Arbeit noch nicht integriert.

Entsprechend dem in den neuropsychologischen Studien verwendeten Stimulusmaterial (siehe Abbildung 2) wurde das Modell an dieser Stelle ausschließlich im Hinblick auf die Detektion von normalen und Navon-Buchstaben analysiert. Dabei wurde das Stimulusmaterial von Huberle and Karnath (2006) verwendet (siehe hierzu auch Abschnitt 4).

Daraus lassen sich bereits einige entscheidende Herausforderungen an die Modellierung ableiten:

- Die Größenunterschiede zwischen globalen und lokalen Elementen innerhalb der Navon-Buchstaben betragen über 1:30. Indem einzelne Bildsegmente selektiv ausgeschnitten und anschließend auf eine Einheitsgröße skaliert werden, lassen sich diese Differenzen zwar reduzieren. Dennoch sollte das Modell bis zu einem gewissen Grade **größeninvariant** sein.
- Zwar wird der globale Buchstabe stets im Bild zentriert präsentiert. Die lokalen Elemente jedoch sind an den verschiedensten Bildpositionen zu finden. Auch dieser Faktor lässt sich durch den Aufmerksamkeitsfilter reduzieren. Da aber nach wie vor mehrere lokale Elemente innerhalb des Ausschnitts sein können, muss das Modell auch **Lokationsinvarianz** umsetzen können.
- Die kleinen Buchstaben sind im Verhältnis wesentlich dicker als der große, globale Buchstabe. Dieses Problem wird durch das Normieren der Bildausschnitte, und damit dem Annähern der Buchstabengrößen, noch verstärkt. Daher sollte auch eine gewisse **Dickeninvarianz** gewährleistet sein.

Im ersten Teil des Projekts wurde daher die Invarianz des Modells gegenüber verschiedenen Buchstabengrößen, -dicken und -positionen innerhalb des Gesamtbildes untersucht.

Während bei der größten und kleinsten Filtergröße jeweils nur ein Buchstabe zu sehen ist, können bei der mittleren Filtergröße auch mehrere Elemente im Bild sein. Daher wurde das Modellverhalten bei Präsentation mehrerer identischer Buchstaben analysiert. Außerdem wurde untersucht, wie das Modell bei der Präsentation verschiedenartiger Buchstaben reagieren würde.

Zum Abschluss wurde das optimierte Modell auf die Navon-Buchstaben getestet und der vereinfachte Aufmerksamkeitsfilter integriert.

### 3 Allgemeine Simulationen

In den folgenden Simulationen wurde das HMAX-Modell mit den in Abschnitt 1.2 genannten Parametereinstellungen verwendet. Die in Mutch and Lowe (2006) beschriebenen Erweiterungen wie Sparse Features und lokale Gebundenheit der Features wurden hier nicht verwendet, da dies der Komplexität des Stimulusmaterials nicht angemessen war. Die Anzahl der zu lernenden Features wurde von uns auf 500 festgelegt. Da das hier verwendete Stimulusmaterial sehr viele homogene Flächen enthält, was zu vielen unbrauchbaren Features führen würde, wurden nur solche in das Feature Dictionary aufgenommen, deren Varianz einen Wert von  $\sigma = 0.0001$  übersteigt.

In Übereinstimmung mit der dort verwendeten Größe des Eingabebildes von 140x140px wurde auch die Normgröße des Aufmerksamkeitsfilters auf diesen Wert gesetzt. Zu beachten ist dabei, dass diese Bildgröße *nicht* der Buchstabengröße entspricht, da um die Buchstaben immer noch etwas Abstand zum Bildrand gelassen wurde. Ein Buchstabe in einem 140x140px großen Bild hat dementsprechend nur eine Größe von 108x108px!

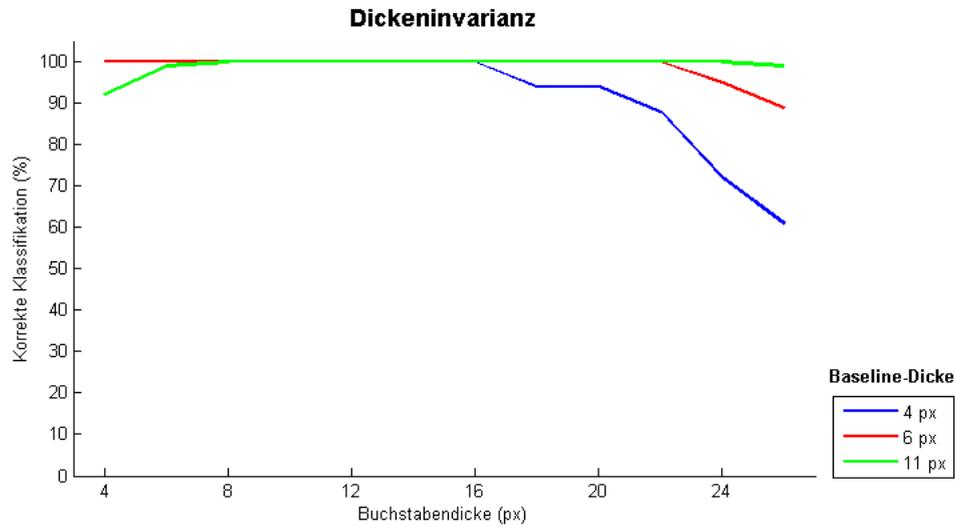
Für die allgemeinen Tests zur Invarianz wurde daher ebenfalls diese Bildgröße verwendet. Alle Simulationen wurden mit mehreren Feature Dictionaries und Trainingsdurchläufen durchgeführt, um die allgemeine Leistungsfähigkeit des Modells zu untersuchen (in den folgenden Abbildungen ist daher stets der Mittelwert angegeben). Basierend auf diesen Ergebnissen wurde anschließend ein gutes Feature Dictionary sowie ein optimierter Klassifikator erstellt, welche für die weiteren Simulationen mit den Navon-Buchstaben verwendet wurden.

#### 3.1 Invarianz gegenüber Größe, Dicke und Position der Buchstaben

**Dickeninvarianz.** In diesem Szenario wurde untersucht, inwieweit das Modell die Dicke eines Buchstabens generalisieren kann, wenn zuvor nur auf eine Dicke (Baseline) trainiert wurde.

Pro Durchgang wurden dazu Features aus dem Baseline-Buchstaben (siehe Abbildung 12) extrahiert und das Modell mit diesem Feature Dictionary trainiert. Getestet wurden anschließend Buchstaben variabler Dicke. Diese wurden automatisch generiert, indem die Ränder des dünnsten Buchstabens sukzessive vergrößert wurden. Alle Elemente wurden einmalig in zentrierter Position und mit einer konstanter Größe von 108x108px präsentiert. Jede der drei Baseline-Größen wurde mit jeweils 20 verschiedenen Feature Dictionaries untersucht.

Wie Abbildung 11 zu entnehmen ist, reagiert das Modell sehr robust gegenüber Variationen der Buchstabendicke. Nur beim Training sehr dünner Buchstaben sinkt die Performanz bei der Klassifikation sehr dicker Buchstaben auf etwa 60% (was aber auch noch deutlich über Chance Level ist).



**Figure 11:** Simulationen zur Dickeninvarianz: Dargestellt ist die Performanz des Modells bei Präsentation unterschiedlich dicker Buchstaben, gemittelt über alle Durchgänge. Als Baseline wurden dabei jeweils die in Abbildung 12 dargestellten Buchstaben verwendet. Selbst bei enormen Unterschieden (blau) hinsichtlich der Dicke der Buchstaben wird immer noch eine passable Performanz von über 60% erreicht.

Da diese Dicke bereits außerhalb der in diesem Kontext verwendeten Buchstabenformate liegt, ist dieser Performanzrückgang an dieser Stelle nicht weiter von Bedeutung.

Im Folgenden wurde daher stets mit dem Buchstaben mittlerer Dicke gearbeitet. Zur Kontrolle wurden gelegentlich stichprobenartige Durchläufe mit den anderen beiden Dicken durchgeführt. In allen Fällen blieb das Ergebnis dabei unverändert.



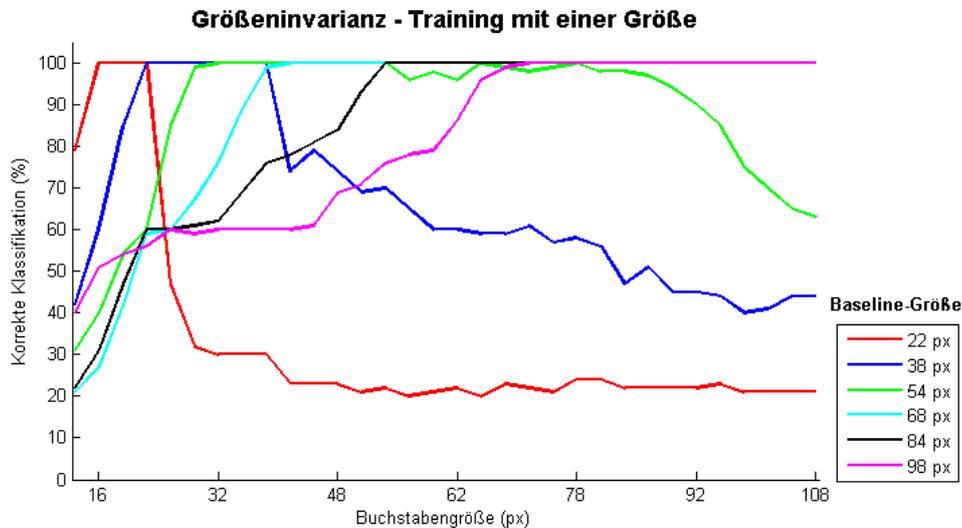
**Figure 12:** In den Simulationen zur Dickeninvarianz wurden die obigen drei Buchstabentypen (mit Dicken von 4, 6 und 11px) verwendet. Das Testmaterial wurde automatisch generiert, indem die Ränder des dünnsten Buchstabens (links) sukzessive vergrößert wurden. Da das Modell diesbezüglich sehr gut generalisieren konnte, wurde anschließend nur noch der mittlere Buchstabe (Mitte) verwendet.

Diese Ergebnisse sind noch in anderer Hinsicht interessant: Durch

das automatische Vergrößern der Buchstabendicke wurden auch die Buchstabenform etwas verändert – beispielsweise ist der Querbalken des dünnen A weiter oben als der des dicken A; zudem sind die Linienenden mit zunehmender Dicke stärker abgerundet. Die Performanz scheint dadurch jedoch nicht beeinträchtigt zu sein. Auch wenn dieser Aspekt an dieser Stelle nicht im Detail untersucht werden soll, so kann dennoch eine gewisse Robustheit gegenüber leichten Formänderungen angenommen werden.

**Größeninvarianz.** In diesem Szenario wurde untersucht, inwieweit das Modell die Größe eines Buchstabens generalisieren kann, wenn zuvor nur auf eine Größe (Baseline) trainiert wurde.

Pro Durchgang wurden dazu wieder Features aus dem Baseline-Buchstaben extrahiert und das Modell mit diesem Feature Dictionary trainiert. Getestet wurden anschließend Buchstaben in der Größenordnung von 14 bis 108px (die Untergrenze wurde dabei etwas kleiner als die Größe der lokalen Elemente in den Navon-Buchstaben gewählt). Alle Elemente wurden einmalig an zentrierter Position und mit im Verhältnis konstanter Dicke präsentiert. Jede der insgesamt 32 Baseline-Größen wurde mit jeweils 40 verschiedenen Feature Dictionaries untersucht.



**Figure 13:** Simulationen zur Größeninvarianz: Das Modell wurde auf eine bestimmte Buchstabengröße trainiert und anschließend mit Buchstabengrößen von 14 bis 108px getestet. Offensichtlich kann mit einem solchen Setting im Mittel nur eine begrenzte Größeninvarianz erreicht werden.

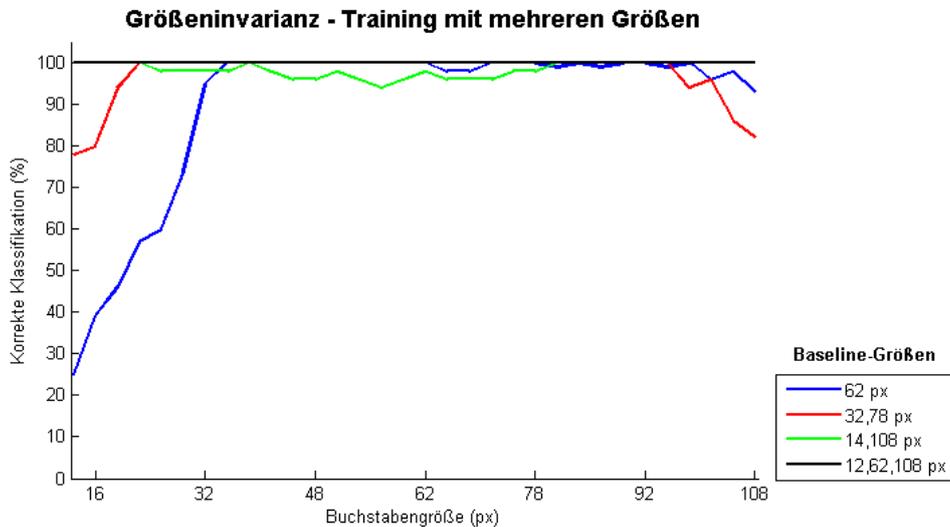
Abbildung 13 zeigt deutlich, dass der Klassifikator unter diesen Bedingungen nur in begrenztem Maße generalisieren kann. Erstaunlicherweise werden in allen Fällen stets auch noch solche Buchstaben erkannt, die bis zu 10px (bei sehr kleinen) oder gar 40px (bei größeren) kleiner als die Baseline-

Buchstaben sind. Eine Generalisierung auf vergleichsweise größere Buchstaben scheint erst ab einer Größe von etwa 54px möglich zu sein.

Diese Ergebnisse werfen im Hinblick auf die Aufmerksamkeitsfilter eine entscheidende Frage auf, da abhängig von deren Auflösung auch die Größe der zu erkennenden Buchstaben variiert. Trainiert man das HMAX-Modell nur auf eine mittlere Buchstabengröße von 62px, so kann bereits ein Größenspektrum von etwa 40 bis 100px abgedeckt werden (blaue Linie in Abbildung 14). Dadurch könnte das Modell zwar einfacher gehalten werden, ein 1:1-Filterausschnitt wäre allerdings nicht erfolgreich klassifizierbar.

Alternativ könnte das Modell daher auch auf mehrere Buchstabengrößen trainiert werden. Abbildung 14 zeigt, dass die Performanz des Modells durch das Training mehrerer Buchstaben deutlich verbessert werden kann. Das gesamte Spektrum von 14 bis 108px kann schließlich erfolgreich klassifiziert werden, wenn Extrem- und Mittelgrößen zusammen präsentiert werden.

Besonders erwähnenswert ist an dieser Stelle, dass im Training zwar zwei Buchstabengrößen verwendet wurden, die Features jedoch ausschließlich aus Buchstaben der Größe 62x62px gezogen wurden. Dies zeigt, dass die Performanz des Modells insbesondere hinsichtlich seiner Generalisierbarkeit maßgeblich von den im Training präsentierten Stimuli abhängt; um ein allgemeines Feature Dictionary zu erstellen, ist aber bereits ein Buchstabe ausreichend.



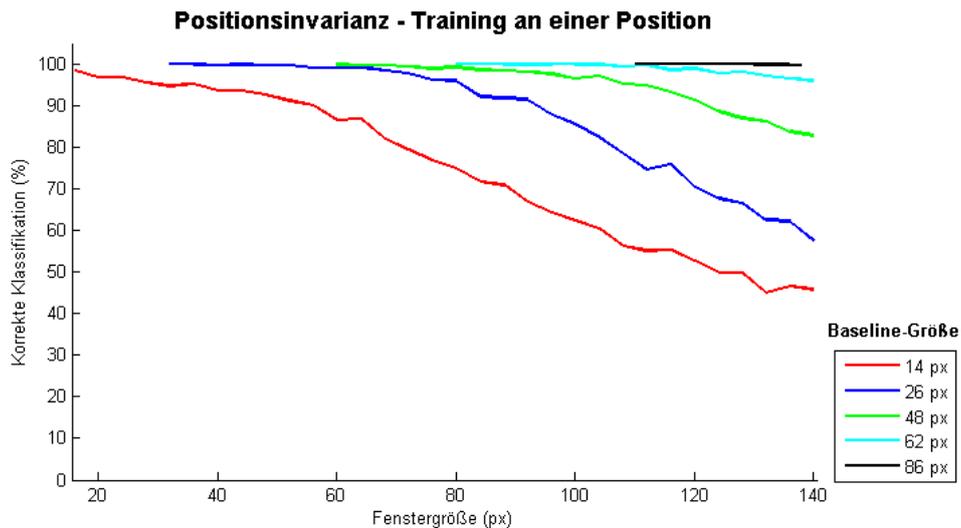
**Figure 14:** Dargestellt ist die Performanz des Modells, gemittelt über alle Durchgänge, beim Training mehrerer Größen. In diesem Fall können auch Zwischengrößen erfolgreich klassifiziert werden. Die besten Ergebnisse erzielt man dabei bei Präsentation der Mittel- und Extremgrößen.

Natürlich gelten diese Ergebnisse (vorerst) ausschließlich für zentriert präsentierte Buchstaben. Die Positionsinvarianz im Allgemeinen sowie im

Zusammenspiel mit obigen Größenvarianzen wird im folgenden Abschnitt untersucht.

**Positionsinvarianz.** In diesem Szenario wurde untersucht, inwieweit das Modell die Position eines Buchstabens generalisieren kann, wenn zuvor nur auf eine Position (Baseline) trainiert wurde.

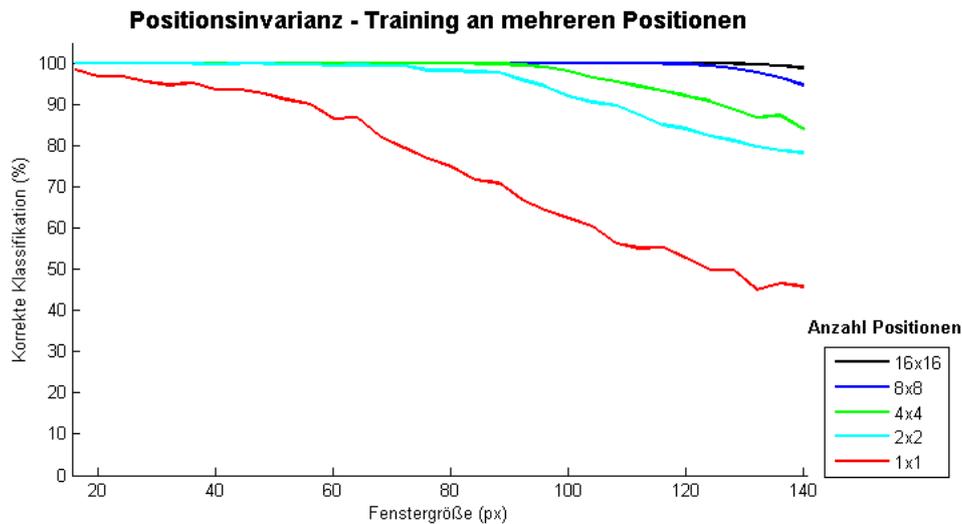
Pro Durchgang wurden dazu Features aus dem zentriert präsentierten Baseline-Buchstaben extrahiert und das Modell mit diesem Feature Dictionary trainiert. Getestet wurden anschließend Buchstaben derselben Größe und Dicke, die an zufälliger Position innerhalb eines bestimmten Fensters um den Baseline-Buchstaben präsentiert wurden. Das kleinste Fenster war dabei stets so groß wie der Baseline-Buchstabe, d.h. Trainings- und Testbuchstabe waren identisch. Beim größten Fenster (140x140px) konnte sich der Buchstabe an jeder beliebigen Stelle des gesamten Bildes befinden. Für jede Fenstergröße wurden die Buchstaben an jeweils 40 verschiedenen Positionen präsentiert.



**Figure 15:** Simulationen zur Positionsinvarianz: Unabhängig von der Buchstabengröße verschlechtert sich die Performanz des Modells graduell mit zunehmender Distanz zwischen trainiertem und getestetem Buchstaben. Auch hier wurde wieder über alle Durchgänge gemittelt.

Abbildung 15 zeigt deutlich, dass Positionsinvarianz mit den verwendeten Parametereinstellungen nur in begrenztem Maße möglich ist. Die Performanz ist stets nahe 100% bei größeren Buchstaben, da bei ihnen aufgrund der begrenzten Bildgröße nur verhältnismäßig geringfügige Positionsinvarianzen möglich sind. Bei kleineren Buchstaben hingegen nimmt die Performanz antiproportional zur Distanz zwischen Trainings- und Testposition ab.

Auch hier könnte man das Modell wieder auf verschiedene Positionen trainieren lassen. Dazu wurde ein 14px großer Buchstabe nacheinander systematisch an verschiedenen Stellen des Gesamtbildes gezeigt, indem dieses in ein  $N \times N$  Felder großes Gitter eingeteilt wurde und die Buchstaben jeweils in der linken oberen Ecke der Felder präsentiert wurden. Die Gittergröße variierte dabei von  $2 \times 2$  Positionen, in diesem Fall werden die Buchstaben im Abstand von etwa vierfacher Buchstabengröße präsentiert, bis zu  $16 \times 16$  Positionen, hier überlappen sich die Buchstabenpositionen bereits.



**Figure 16:** Präsentiert man die Buchstaben im Training an verschiedenen Positionen, so kann die Performanz im Test proportional zur Anzahl der Trainingspositionen gesteigert werden.

Analog zur Größeninvarianz kann auch die Positionsinvarianz deutlich gesteigert werden, wenn im Training mehrere Positionen trainiert werden (siehe Abbildung 16).

### 3.2 Finales Setting

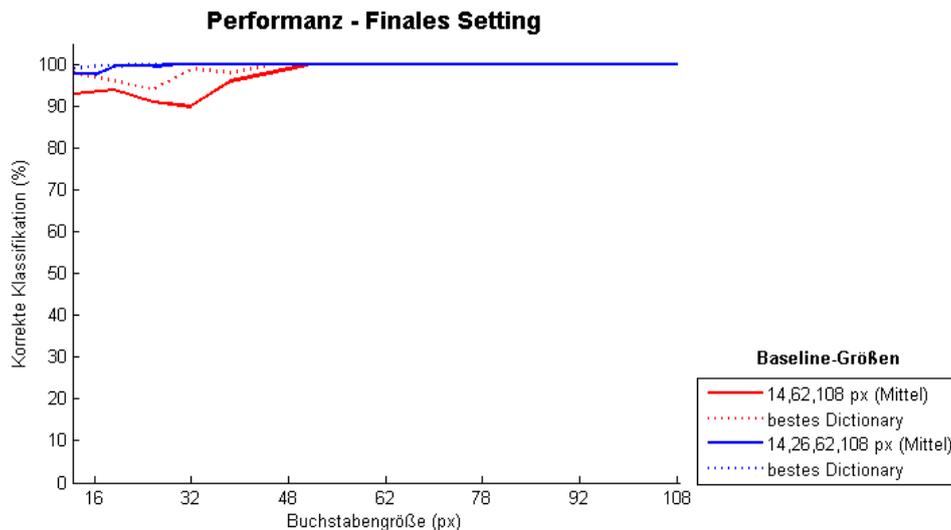
Aus den obigen Simulationen konnten einige wichtige Erkenntnisse hinsichtlich entscheidender Parametereinstellungen gewonnen werden:

1. Ein Buchstabe mittlerer Dicke und Größe ist ausreichend, um ein gutes **Feature Dictionary** zu erstellen, das sich auch auf Buchstaben anderer Größen und Dicken anwenden lässt.
2. Die Performanz des Modells hinsichtlich der Generalisierbarkeit hängt im Wesentlichen von den **im Training präsentierten Stimuli** ab.
3. Das Modell kann sehr gut mit verschiedenen Dicken umgehen, aber nur in begrenztem Maße über verschiedene Größen und Positionen

generalisieren. Dieses Problem kann gelöst werden, indem im Training mehrere Größen und Positionen verwendet werden.

Im Folgenden wurden die Features aus einem 62x62px großen, zentriert präsentierten Buchstaben mittlerer Dicke ermittelt. Anschließend wurde die Support Vector Machine auf die drei Buchstabengrößen 14,62,108px trainiert, die bereits bei der Größenerkennung sehr erfolgreich waren (vergleiche Abbildung 14). Dabei wurde der größte Buchstabe zentriert präsentiert (bei dieser Größe ist keine Positionsvariation möglich), die beiden anderen Größen wurden nacheinander an verschiedenen Positionen gezeigt, indem das Gesamtbild analog zu oben in 10x10 (bei kleinen Buchstaben) bzw. 2x2 Felder (bei mittleren Buchstaben) große Gitter eingeteilt wurde.

Das Modell wurde anschließend noch einmal auf Größen- und Positionsinvarianz getestet. Dazu wurden wieder Buchstaben der Größe 14 bis 108px verwendet, die jeweils an 40 zufälligen Positionen innerhalb des gesamten Bildes gezeigt wurden.



**Figure 17:** Ein auf vier Größen trainierter Klassifikator ist in der Lage, beliebig große Buchstaben, die an zufälligen Positionen präsentiert werden, korrekt zu klassifizieren.

Überraschenderweise war selbst der beste der in 20 Durchgängen ermittelten Klassifikatoren (rot gestrichelte Linie in Abbildung 17) nicht in der Lage, zufällig angeordnete Buchstaben bis Größe 48px mit einer Genauigkeit von über 95% korrekt zu klassifizieren.

Daher wurde im Training zusätzlich noch eine vierte Größe (26px) verwendet, um diesem Performanzrückgang entgegenwirken zu können. Abbildung 17 zeigt, dass der beste der neuen Klassifikatoren (blau gestrichelte Linie) das gesamte Größen- und Positionsspektrum erfolgreich abdecken

kann. Dieser wurde daher in den folgenden Simulationen verwendet.

### 3.3 Modellverhalten bei Präsentation mehrerer Buchstaben

Bisher wurde dem Klassifikator sowohl im Training als auch im Test immer nur ein Buchstabe präsentiert. Beim Auswählen verschiedener Bildsegmente durch einen Aufmerksamkeitsfilter mittlerer Größe ist es allerdings wahrscheinlich, dass sich in diesem Ausschnitt mehrere Buchstaben befinden. Im zweiten Teil des Projektes wurde daher untersucht, inwieweit das Modell (mit dem optimierten Feature Dictionary und Klassifikator aus Abschnitt 3.2) mit mehreren Elementen umgehen kann.

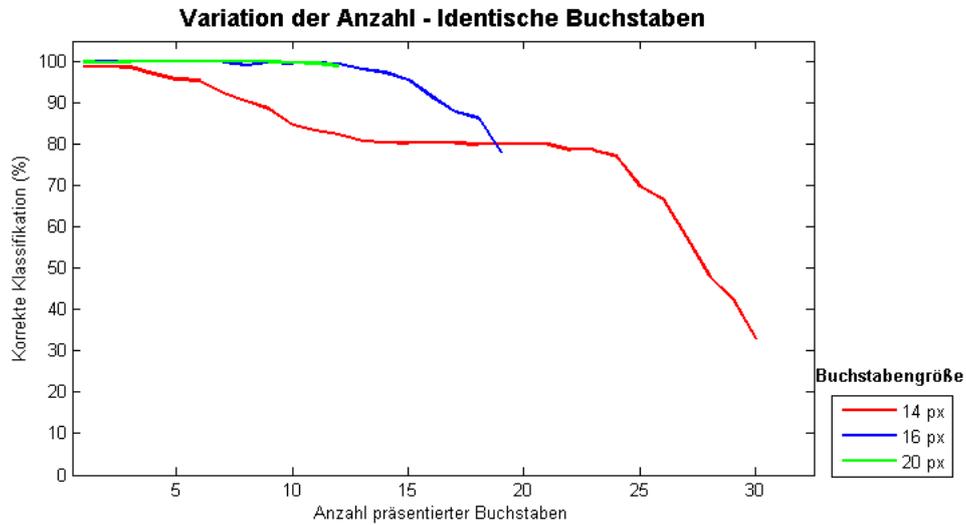
**Identische Buchstaben.** Zuerst wurde analysiert, ob die Performanz des Modells durch Präsentation mehrerer Buchstaben beeinträchtigt werden kann.

Dazu wurden 14, 16 und 20px große Buchstaben verwendet. Das Modell wurde mit jeweils 100 Bildern getestet, auf denen je ein bis 30 (sich nicht überlappende) kleine Buchstaben an zufälligen Positionen des 140x140px großen Gesamtbildes zu sehen waren. Bei den größeren wurde die Anzahl auf 19 bzw. 12 beschränkt, da die zufällige Erstellung entsprechender Bilder andernfalls zu lange gedauert hätte.

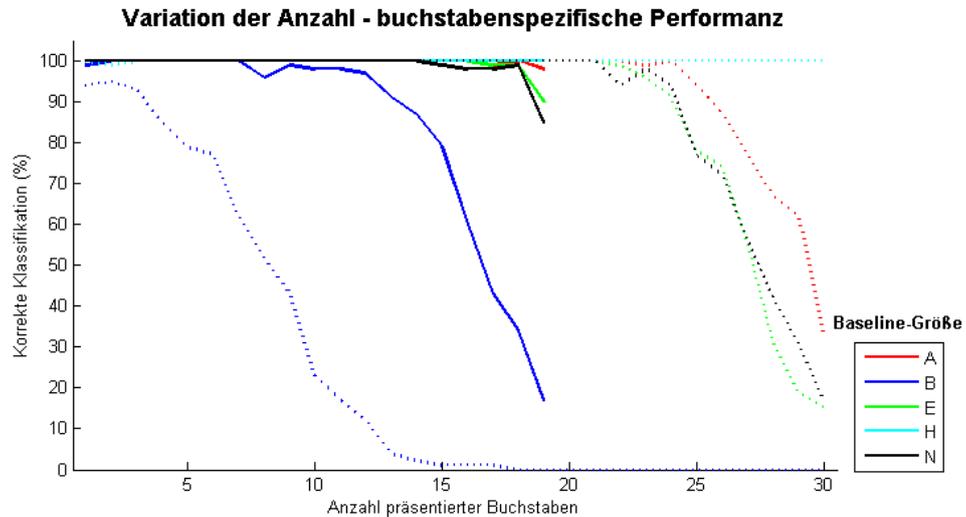
Abbildung 18 zeigt, dass bei den größeren Buchstaben bis zu 15 Elemente erfolgreich erkannt werden können. Erst dann fällt die Performanz deutlich ab. Die Erkennungsleistung bei mehreren kleinen Buchstaben fällt hingegen von Beginn an schnell ab und pendelt sich dann bei 10 bis 20 Elementen auf fast konstant 80% ein, bevor sie noch weiter fällt.

Um mögliche Ursachen für diesen schnellen Performanzrückgang bei kleinen Buchstaben zu finden, wurden die Erkennungsleistungen separat für die einzelnen Buchstaben betrachtet (siehe Abbildung 19). Erstaunlicherweise wird nur das B schlecht erkannt – sowohl bei den kleinen (wodurch die Performanz von 80% zustande kommt) als auch bei den mittleren Buchstaben. Dies könnte möglicherweise daran liegen, dass einige spezifische B-Features (beispielsweise die runden!) aufgrund der geringen Buchstabengröße nicht zum Einsatz kommen können – oder aber dass es zu viele dem E oder H ähnliche Features gibt. Im "Zweifelsfall" entscheidet sich dieses Modell dann stets für das H (welches aus diesem Grund auch permanent 100% erreicht).

Dieses Ergebnis ist im Hinblick auf die Größe des mittleren Filters entscheidend: Befinden sich meist sehr viele Buchstaben im Bildausschnitt, sollte der Filter mit den kleinen Buchstaben arbeiten, da diese bei hohen Anzahlen wesentlich länger und besser erkannt werden als mittlere Buchstaben (in diesem Fall sollte man eventuell sämtliche Navon-Buchstaben mit lokalen Bs aus dem Stimulus-Set herausnehmen).



**Figure 18:** Bis zu 15 größere Buchstaben können fast immer fehlerfrei klassifiziert werden. Bei kleinen Buchstaben pendelt sich die Performanz schnell bei 80% ein, bevor sie ab 20 Elementen weiter sinkt.

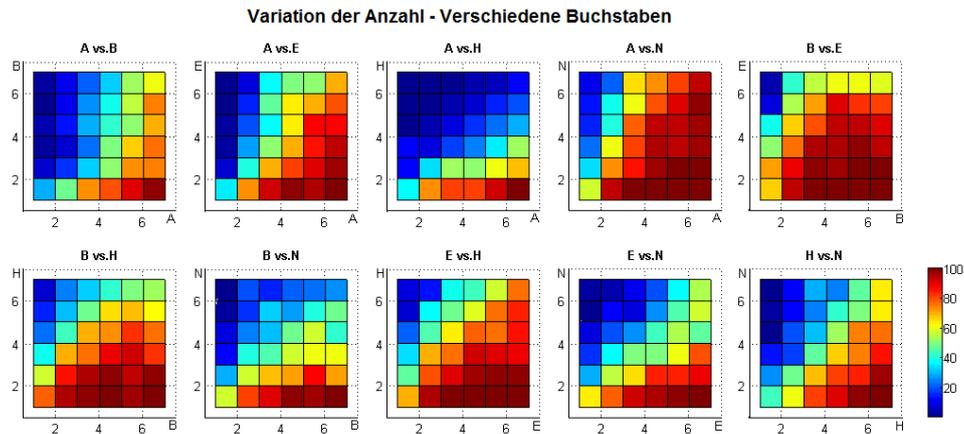


**Figure 19:** Sowohl bei 14px (gestrichelt) als auch bei 16px großen Buchstaben (durchgezogene Linien) können alle Buchstaben sehr gut erkannt werden – bis auf das B.

Da sich aber selbst bei einem 1:1-Ausschnitt aus einem Navon-Buchstaben mit Abstand 5 kaum mehr als 10 Elemente im Bildsegment befinden, sollten die 16px großen Buchstaben gewählt werden, da bei dieser Größe insbesondere auch die Bs noch sehr gut erkannt werden können.

**Verschiedene Buchstaben.** Besonders interessant ist natürlich die Frage nach dem Modellverhalten, wenn verschiedene Buchstaben gleichzeitig präsentiert werden. Intuitiv würde man hierbei erwarten, dass sich das Modell für denjenigen Buchstaben entscheidet, der häufiger im Bild zu sehen ist.

Um diese Hypothese zu untersuchen, wurden im Folgenden jeweils Buchstaben zweier verschiedener Arten gemeinsam präsentiert, wobei die Anzahl jeweils von 1 bis 7 Elemente pro Buchstabenart variierte. Getestet wurden zudem alle 10 Buchstabenkombinationen, so dass sich insgesamt  $7 \times 7 \times 10 = 490$  verschiedene Test-Settings ergaben. Jede dieser Kombination wurde mit jeweils 100 Bildern (140x140px) getestet, auf denen die 16px großen Buchstaben analog zu oben zufällig verteilt waren.



**Figure 20:** Dargestellt ist die Entscheidungshäufigkeit(%) für den zuerst genannten Buchstaben, wenn mehrere Elemente verschiedener Art gemeinsam präsentiert werden (die Anzahl ist an der x- und y-Achse aufgetragen). Offensichtlich entscheidet sich das Modell im Allgemeinen für den Buchstaben, der häufiger vorkommt.

Entsprechend der Hypothese entscheidet sich das Modell im Allgemeinen für den Buchstaben, der häufiger im Bild zu sehen ist (siehe Abbildung 20). Dies ist aber keine Alles-oder-Nichts-Entscheidung, sondern die Performanz steigt proportional zum Mengenverhältnis beider Buchstaben.

Offensichtlich gibt es hier einige dominantere Buchstaben, die selbst bei vergleichsweise geringerer Anzahl die Modellvorhersage bestimmen. Besonders deutlich wird dies bei den Tests "A vs. H", "B vs. E" und "B vs. H". Die Dominanz des H im ersten Fall könnte man sich – wie oben erwähnt – dadurch erklären, dass sich das Modell im Zweifelsfall stets für das H entscheidet. Überraschenderweise dominiert in den anderen beiden Fällen aber das B – also genau der Buchstabe, der in den vorhergehenden Simulationen stets am schlechtesten erkannt wurde.

An dieser Stelle werden diese Effekte nicht näher analysiert. In weit-

erführenden Studien sollten sie aber genauer analysiert werden, wobei man insbesondere auch die Ähnlichkeit zweier Buchstaben systematisch untersuchen sollte.

## 4 Anwendung des Modells auf Navon-Buchstaben

Im letzten Abschnitt wurde das Modell nun mit den Navon-Buchstaben untersucht. Das verwendete Navon-Set bestand dabei aus insgesamt 100 Bildern: Fünf globale Buchstaben (A,B,E,H,N), erstellt aus jeweils einem der vier übrigen Buchstabentypen mit fünf verschiedenen Inter-Element-Abständen (siehe hierzu Abbildung 2).

Um diese wurde zunächst ein etwa 50px breiter weißer Rand hinzugefügt, um sie dem oben verwendeten Material anzugleichen. Proportionsunterschiede blieben allerdings bestehen (so ist beispielsweise das Navon-A nicht nur breiter als das normale A, sondern auch kleiner als die übrigen Navon-Buchstaben!).

### 4.1 Vereinfachter Aufmerksamkeitsfilter

Im Hinblick auf die Gestaltung des Aufmerksamkeitsfilters (siehe hierzu auch Abschnitt 2.1) stellten sich vor allem zwei Fragen:

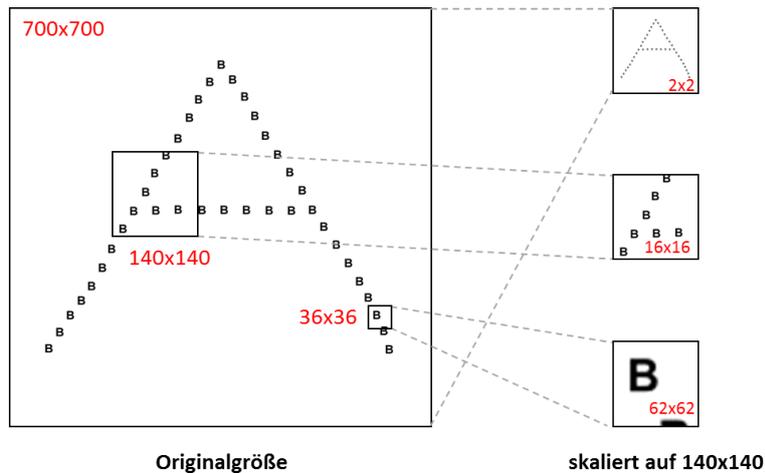
1) *Wie wird ein lokaler bzw. globaler Aufmerksamkeitsfokus umgesetzt?* In der vorliegenden Arbeit sollen mit unterschiedlich großen Filtern Bildsegmente aus dem Gesamtbild ausgeschnitten werden, wodurch – je nach Filtergröße – eher lokale bzw. globale Elemente in den Aufmerksamkeitsfokus rücken (siehe Abbildung 21).

Im Folgenden werden dabei drei Größen zum Einsatz kommen: Zur Erkennung des globalen Buchstaben ist es unumgänglich, das gesamte Bild (skaliert auf 140x140px) dem Modell zu übergeben – andernfalls wären nur Buchstabenfragmente enthalten, was die Identifikation unmöglich machen würde. Weil die lokalen Elemente durch das Normieren auf 2x2px reduziert werden, liegt in diesem Fall der Fokus auch ausschließlich auf den globalen Buchstaben. Da das Modell 16px große Buchstaben noch sehr gut klassifizieren kann, ist auch ein 1:1-Ausschnitt von 140x140px problemlos möglich. Der kleinste Filter sollte ausschließlich auf die lokalen Buchstaben fokussieren. Hier wurde eine Filtergröße von 36x36px gewählt, so dass dem Modell Buchstaben mittlerer Größe (62px) übergeben werden.

2) *Nach welchem Prinzip werden diese Bildsegmente ausgewählt?* Beim großen Filter wird dem Modell der gesamte Navon-Buchstabe, skaliert auf 140x140px, übergeben. Da sich diese Stimuli deutlich vom bisher verwendeten Stimulusmaterial unterscheiden, werden sie im Folgenden noch einmal gesondert untersucht.

Mit den beiden kleineren Filtern werden nun einzelne Bildpatches aus dem Navon-Buchstaben ausgeschnitten. Dafür wurde das Gesamtbild mit dem Filter systematisch abgescannt, indem der Filter in Abständen von 35px (beim mittleren) bzw. 9px (beim kleinen Filter) über das Bild bewegt wurde und die Grauwerte an jeder Position aufsummiert wurden. Von den so analysierten Bildpatches wurden jeweils die 10 (beim mittleren) bzw. 20 (beim kleinen Filter) Segmente mit den höchsten Werten ausgewählt und dem Klassifikator übergeben. Dieses Auswahlprinzip könnte man bereits als rudimentären bottom-up Prior betrachten.

Der Filter wählt also solche Ausschnitte aus, in denen möglichst viele schwarze Pixel zu finden sind, wodurch automatisch die Anzahl der enthaltenen Buchstaben maximiert wird. Dies sind im Allgemeinen die Bereiche innerhalb der Buchstaben, an denen mehrere Kanten zusammentreffen. Dabei kann es allerdings vereinzelt vorkommen, dass einige Elemente nur anteilig zu sehen sind – was die Performanz eventuell beeinträchtigen könnte. Im Allgemeinen unterscheiden sich die gefundenen Bildausschnitte aber nur geringfügig von dem in den vorangegangenen Simulationen verwendeten Material.



**Figure 21:** Umsetzung des Aufmerksamkeitsfilters: Je nachdem ob der Fokus eher auf lokalen oder globalen Bildelementen liegt, wird ein kleineres oder größeres Bildsegment ausgewählt. Dieses wird normiert und anschließend dem HMAX-Modell übergeben. Dadurch wird statt der ursprünglichen Buchstabengrößen von 600 und 16px ein gemäßigeres Größenspektrum von 108,62 und 16px erreicht.

## 4.2 Parameteranpassungen bei Verwendung von Navon-Buchstaben

Als erstes wurde untersucht, ob das Modell mit den Parametereinstellungen aus Abschnitt 3.2 in der Lage ist, die Navon-Buchstaben zu klassifizieren.

Hierbei wurde stets das vollständige Navon-Set (im Folgenden entsprechend der in Abbildung 2 genannten Abstände als Navon-1 bis Navon-5 bezeichnet) und zur Kontrolle auch weiterhin normale Buchstaben verschiedener Größen und Positionen getestet.

Abbildung 22 (dunkelblaue Balken) zeigt, dass mit diesem Modell kein einziger Navon-Buchstabe korrekt erkannt werden kann: Selbst bei den Navon-5-Buchstaben übersteigt die Performanz nur knapp das Chance Level. Das Modell ist also nicht in der Lage, von normalen auf Navon-Buchstaben zu generalisieren, wenn diese nicht trainiert wurden.

Eine mögliche Erklärung für dieses schlechte Ergebnis wäre, dass die gespeicherten Features (die ja aus einem normalen Buchstaben extrahiert wurden!) nicht hinreichend genau mit denen der Navon-Buchstaben übereinstimmen. Um dies zu überprüfen, wurde das Feature Dictionary beibehalten und der Klassifikator ausschließlich auf Navon-Buchstaben trainiert. Um Overfitting zu vermeiden, sollte dabei optimalerweise nur ein Navon-Buchstabe jeder Klasse gezeigt werden. Diesbezüglich stellt sich nun die Frage, *welche* Navon-Buchstaben verwendet werden sollten.

Im Hinblick auf den Inter-Element-Abstand wären zwei Ansätze vorstellbar: Einerseits könnte man Navon-5-Buchstaben im Training verwenden, um die Unterschiede zu den normalen Buchstaben möglichst gering zu halten. Alternativ könnten auch Navon-3-Buchstaben gezeigt werden, da diese Generalisierungsspielraum sowohl auf größere als auch auf kleinere Abstände zulassen. Da beide Ansätze plausibel erscheinen, wurden sie im Folgenden parallel untersucht (hierbei werden die Bezeichnungen N3- bzw. N5-Klassifikator verwendet werden).

Vorstellbar wäre außerdem, dass verschiedene Buchstaben (des selben Abstands) unterschiedlich gut für das Training geeignet sind: Ein (globales) A aus lokalen Bs könnte eventuell besser geeignet sein als eines aus lokalen Ns. Um die besten Buchstaben zu finden, wurden alle  $4^5$  Kombinationen nacheinander trainiert und anschließend jeweils das komplette Navon-Set getestet.

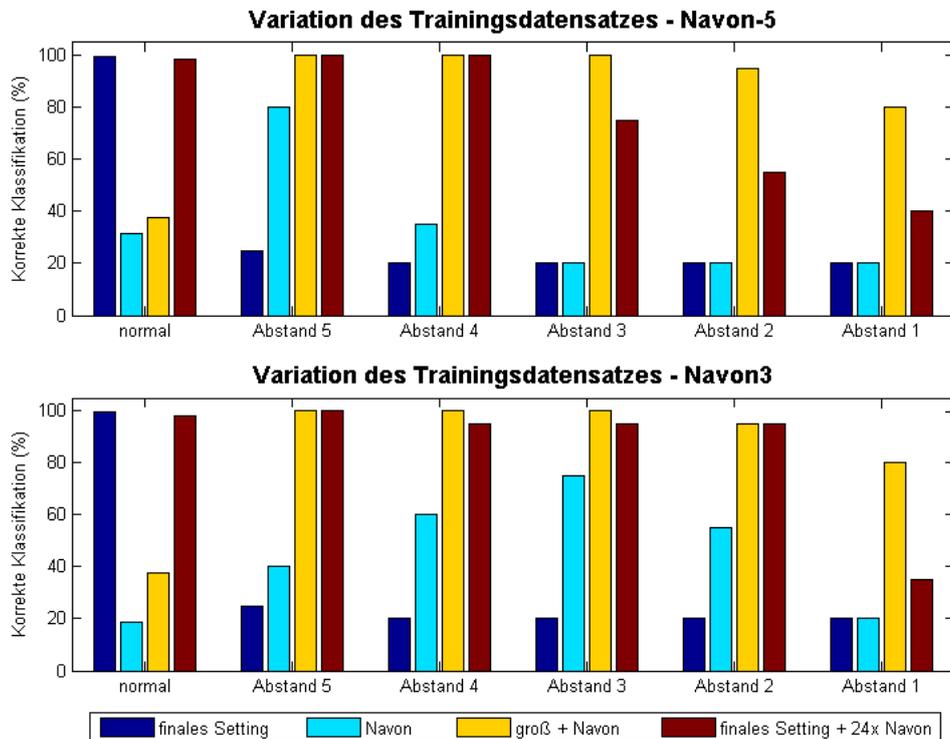
In diesem Fall (hellblaue Balken in Abbildung 22) werden die Navon-Buchstaben deutlich besser erkannt (aber immer noch verhältnismäßig schlecht) – insbesondere wenn eine gute Buchstabenkombination im Training verwendet wird –, normale Buchstaben hingegen werden fast gar nicht mehr korrekt klassifiziert. (Als beste Kombinationen erwiesen sich dabei für das N3-Modell die Buchstaben AausE, BausN, EausB, HausA, NausH sowie im N5-Modell die Buchstaben AausB, BausH, EausH, HausE, NausE. Was diese Kombinationen jetzt gegenüber den übrigen, teilweise gravierend schlechteren auszeichnete, wurde an dieser Stelle nicht näher untersucht.)

Außerdem bestätigten sich die Vermutungen zum Modellverhalten beim Training auf Navon-5- (oben) bzw. Navon-3-Buchstaben (unten): Erwartungsgemäß wurden die Buchstaben mit gleichem Inter-Stimulus-

Abstand wie das Trainingsmaterial am besten klassifiziert. Während das N3-Modell tatsächlich auch auf benachbarte Abstände generalisieren konnte, erkannte das N5-Modell die normalen Buchstaben aufgrund der Ähnlichkeit zum Trainingsmaterial weiterhin besser.

Alles in allem kann aus diesem Ergebnis geschlussfolgert werden, dass das Feature Dictionary zwar durchaus ausreichend ist, um auch Navon-Buchstaben zu klassifizieren. Aber auch hier gilt wieder, dass nicht (bzw. nur wenig) auf normale Buchstaben generalisiert werden kann, wenn diese nicht trainiert wurden.

Als Kompromiss könnte der Trainingsdatensatz daher um Navon-Buchstaben erweitert werden. Dies würde gewissermaßen zusätzliche top-down Informationen darstellen.



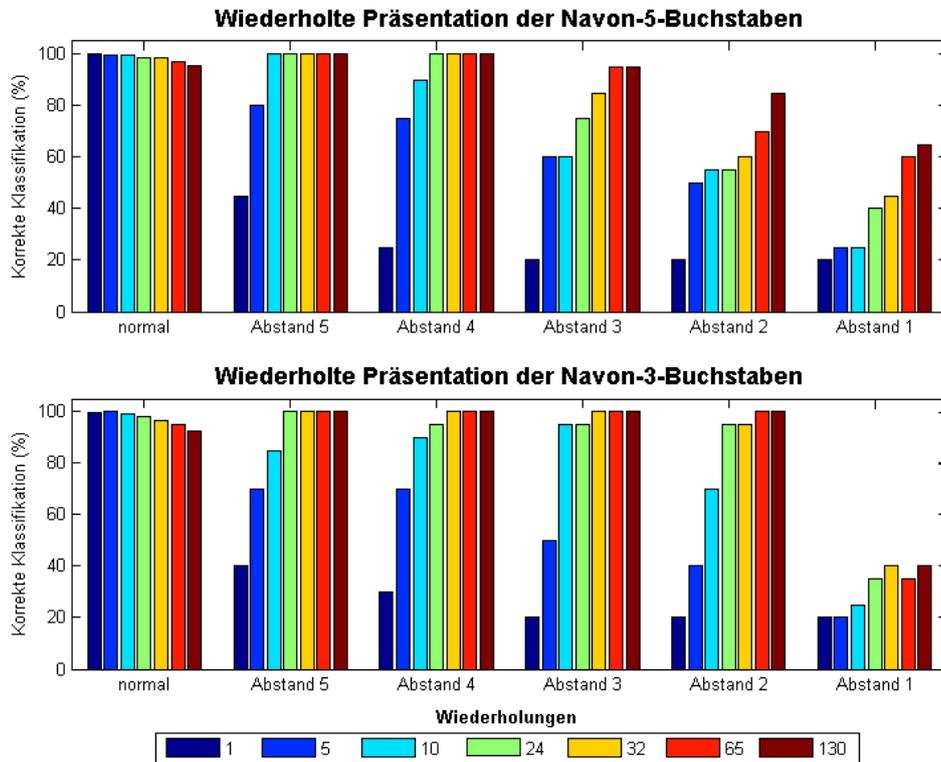
**Figure 22:** Offensichtlich werden nur solche Buchstaben (Navon vs. normal) erkannt, welche auch trainiert wurden. Verwendet man daher beides im Training, steigt die Gesamtperformanz deutlich an.

In einer ersten Simulation wurden dabei die besten Navon-Buchstaben sowie normale 108px-Buchstaben, die jeweils einmalig zentriert präsentiert wurden, als Trainingsstimuli verwendet.

Erstaunlicherweise konnte allein dadurch die Identifikationsleistung bei

Navon-Buchstaben immens gesteigert werden (gelbe Balken in Abbildung 22): Selbst bei Navon-1-Buchstaben wurden um die 80% korrekt klassifiziert, bei den dichteren Abständen lag die Performanz stets nahe 100% (die schlechte Erkennung der normalen Buchstaben ist wieder auf die fehlende Generalisierbarkeit über alle Größen zurückzuführen).

Möglicherweise wirkt der große Buchstabe dabei als eine Art "Verstärker", indem er dem Modell gewissermaßen wichtige Features "zeigt", die bei ausschließlicher Präsentation der Navon-Buchstaben zwar vorhanden sind, aber als zu schwach bzw. zu unaussagekräftig angesehen wurden. Neben der Priorität einzelner Features wird quasi auch ein Spektrum möglicher Intensitäten dieses Features gelernt. Mit dieser Theorie könnte auch die schlechte Performanz im ursprünglichen Modell erklärt werden: Da der Klassifikator stets nur mit starken Features gearbeitet hat, werden die schwächeren Features der Navon-Buchstaben (resultierend aus den unvollständigen Kanteninformationen) nicht als solche wahrgenommen und diese daher falsch klassifiziert. Dieser Aspekt sollte in weiterführenden Studien näher untersucht werden.



**Figure 23:** Die Performanz des Modells lässt sich deutlich steigern, wenn die Navon-Buchstaben im Training wiederholt präsentiert werden.

Da für die Aufmerksamkeitsfilter alle Buchstabengrößen erkannt werden müssen, wurden in einer zweiten Simulation wieder der vollständige Trainingsdatensatz sowie die besten Navon-Buchstaben verwendet.

Dabei wurde gleichzeitig untersucht, ob sich die Performanz steigern lässt, wenn der Navon-Buchstabe im Training wiederholt präsentiert wird. Dies beruhte auf der Überlegung, dass beispielsweise ein Navon-A nicht als "typisches" A angesehen werden könnte, da es als solches nur einmal im Training auftaucht – neben 130(!) normalen As (diese hohe Zahl resultiert aus den zahlreichen Trainingspositionen für die kleineren Buchstaben). Präsentiert man das Navon-A also häufiger im Training, könnte es ebenso zum typischen A werden und die Performanz steigen. Im Folgenden wurden die Navon-Buchstaben daher jeweils 1,5,10,24,32,65 oder 130 Mal wiederholt präsentiert – das Modell sah also nicht verschiedene Navon-Buchstaben einer Klasse, sondern jedes Mal den gleichen!

Die Hypothese konnte bestätigt werden (siehe Abbildung 23): Je häufiger die Navon-Buchstaben im Training gezeigt wurde, desto besser konnten sie anschließend im Test identifiziert werden. Dieser Effekt zeigte sich in beiden Modellen und bei allen Abständen. Andererseits nahm die Performanz bei normalen Buchstaben antiproportional dazu ab. Dieser Trade-Off scheint im Mittel zwar relativ gering zu sein, aber insbesondere für mittelgroße Buchstaben sinkt die Performanz teilweise auf bis zu unter 80%.

In der abschließenden Simulation wurden die Navon-Buchstaben daher 24-mal wiederholt (braune Balken in Abbildung 22). Mit diesem Setting können sowohl normale Buchstaben aller Größen und Positionen als auch die Navon-Buchstaben sehr gut klassifiziert werden – wobei auch hier wieder die Performanz entsprechend der Trainingsstimuli variiert: Im N5-Modell werden die Navon-5 und Navon-4 stets korrekt identifiziert; im N3-Modell werden sogar bis Navon-2 Spitzenwerte erreicht.

### 4.3 Simulationen mit integriertem Aufmerksamkeitsfilter

Zum Schluss wurden die eingangs beschriebenen Aufmerksamkeitsfilter in das Modell integriert. Der Übersichtlichkeit halber soll an dieser Stelle noch einmal das gesamte Design beschrieben werden:

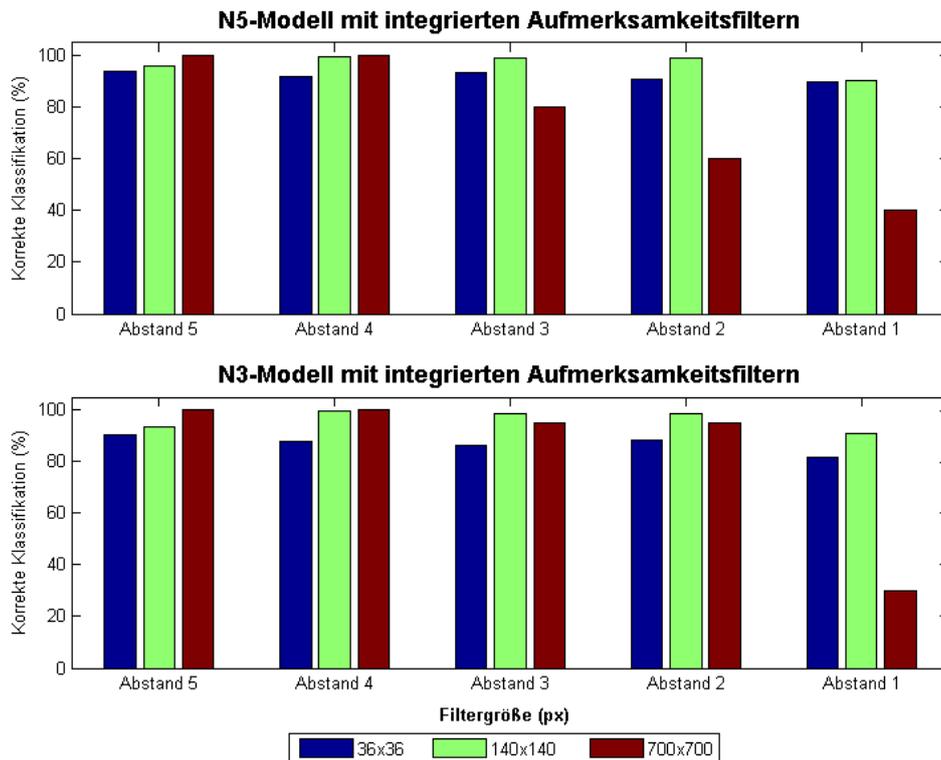
1. **Feature Dictionary:** Die Navon-Buchstaben sind aus den fünf Buchstaben(klassen) A,B,E,H,N zusammengesetzt. Für jede dieser Klassen wurden aus einem zentriert präsentierten, 62x62 px großen Buchstaben mittlerer Dicke je 100 Features extrahiert. Das gesamte Feature Dictionary umfasste also 500 Features.
2. **Klassifikator-Training:** Um Navon-Buchstaben sowie normale Buchstaben variabler Größe und Position erkennen zu können, wurde der Klassifikator auf folgende Stimuli trainiert (vergleiche auch Abschnitt 3.2): 14px große Buchstaben an 100 verschiedenen Positio-

nen (durch Einteilung des Gesamtbildes in ein 10x10 Gitter), 26px-Buchstaben an 25 Positionen, 62px-Buchstaben an 4 Positionen und ein 108px-Buchstabe in Bildzentrum. Außerdem der 108px große beste Navon-Buchstabe, der 24-mal wiederholt im Bildzentrum gezeigt wurde.

Das gesamte Trainingsmaterial beinhaltet damit 770 Stimuli (130 normal + 24 Navon x 5 Klassen).

3. **Klassifikator-Test:** Getestet wurden abschließend Bildausschnitte aus den 100 Navon-Buchstaben (4 lokale Elemente x 5 Klassen x 5 Abstände), die durch die Aufmerksamkeitsfilter generiert wurden. Der 700x700px große Filter (gesamtes Bild) wurde dabei jeweils einmal angewendet, der 140x140px-Filter je 10-mal und der 36x36px-Filter je 20-mal.

Das gesamte Testmaterial umfasste damit 3100 Stimuli (1+10+20 Ausschnitte x 100 Navon).



**Figure 24:** Schneidet man aus den Navon-Buchstaben einzelne Bildsegmente aus (blaue und grüne Balken), so können diese im Allgemeinen sehr gut erkannt werden. Ein Performanzrückgang ist nur beim größten Filter (rote Balken) zu beobachten, wenn die lokalen Elemente zu weit auseinander liegen.

Die Ergebnisse dieser Simulation stimmen im Wesentlichen mit den in den Vorstudien gewonnenen Erkenntnissen überein:

Die Performanz bei Anwendung des größten Filters ist erwartungsgemäß analog zu der in Abbildung 22 dargestellten Performanz des finalen Modells (braune Balken): Buchstaben mit sehr engem Inter-Element-Abstand werden in beiden Modellen zu 100% korrekt klassifiziert. Im N5-Modell nimmt die Identifikationsleistung ab Navon-3-Buchstaben mit zunehmendem Inter-Element-Abstand deutlich ab, während dieser Performanzrückgang im N3-Modell erst beim Navon-1 beobachtet wird.

Bei den beiden kleineren Filtern befinden sich stets nur ein 62px- oder mehrere 16px-Buchstaben im Bildausschnitt. Entsprechend den Ergebnissen zur Größeninvarianz und zur Performanz bei Präsentation mehrerer identischer Buchstaben sollten diese Bilder – unabhängig vom Inter-Element-Abstand – stets korrekt identifiziert werden können. Diese Hypothese wurde im Allgemeinen bestätigt (blaue und grüne Balken in Abbildung 24). Die Gesamtperformanz ist in dieser Simulation allerdings insbesondere beim kleinsten Filter deutlich verringert; hier sinkt die Identifikationsleistung in beiden Modellen auf knapp 90%. Dies könnte vor allem durch den in Abschnitt 4.1 beschriebenen Auswahlmechanismus erklärt werden. Da bei diesem Algorithmus ausschließlich "Pixel gezählt" werden, nicht aber überprüft wird, ob es sich dabei um vollständige Buchstaben handelt, kann es vor allem bei sehr engem Inter-Element-Abstand dazu kommen, dass gewissermaßen zwei halbe Buchstaben im Bildausschnitt enthalten sind – was die Performanz beeinträchtigt.

Insgesamt ist die Performanz des gesamten Modells aber überaus zufriedenstellend.

## 5 Abschließende Anmerkungen

In der vorliegenden Arbeit wurde das biologische motivierte HMAX-Modell zur visuellen Objekterkennung von Serre et al. (2007) auf die von Huberle and Karnath (2006) verwendeten Navon-Stimuli angewendet, um allgemeine Grenzen und Möglichkeiten des Modells im Hinblick auf die Erkennung derartigen Stimulusmaterials auszuloten. Verschiedene Simulationen führten dabei zu folgenden wichtigen Erkenntnissen:

- Die HMAX-Architektur ist ein sehr mächtiges Modell, mit dem sich auch schwierige Aspekte des verwendeten Stimulusmaterials (beispielsweise enorme Größen- und Positionsunterschiede) sehr gut modellieren lassen. Dabei ist die Auswahl eines geeigneten Feature Dictionaries sowie eines mächtigen Klassifikators von zentraler Bedeutung.
- Ein Feature Dictionary, welches aus einem einzigen (normalen) Buchstaben je Klasse erstellt wurde, ist ausreichend, um das gesamte Stim-

ulusmaterial – einschließlich der Navon-Buchstaben – beschreiben zu können.

- Um eine große Bandbreite an Buchstabengrößen beliebiger Positionen im Test abdecken zu können, reicht es, den Klassifikator auf verhältnismäßig wenige Elemente zu trainieren. Sollen zudem Navon-Buchstaben hinreichend gut identifiziert werden können, ist eine wiederholte Präsentation im Training erforderlich.
- Die Integration einfacher Aufmerksamkeitsfilter ermöglicht die Fokussierung auf lokale oder globale Bildelemente. Beim Fokus auf globale Elemente sinkt die Performanz im Allgemeinen mit zunehmendem Inter-Element-Abstand; bei Fokussierung lokaler Elemente hingegen ist die Performanz unabhängig von Inter-Element-Abstand.

Diese Ergebnisse sind insbesondere dahingehend vielversprechend, als dass sie bereits das bei den Simultanagnosie-Patienten beobachtete Antwortverhalten bei Präsentation ebendieser Stimulusmaterials simulieren (vergleiche Abbildung 3 und Abbildung 24, braune Balken).

Zwar erreichten gesunde Kontrollprobanden in den Experimenten von Huberle and Karnath (2006) stets 100%, dennoch ist nicht ganz auszuschließen, dass auch sie ein ähnliches Antwortmuster aufzeigen würden, wäre sämtliches Vorwissen ausgeschaltet. Denn offensichtlich greifen bei ihnen verschiedene Top-Down Verarbeitungsmechanismen (wie sie beispielsweise auch bei der Kanizsa-Illusion (Kanizsa, 1955) zu beobachten sind), welche sie eine geschlossene, zusammenhängende Gestalt wahrnehmen lassen – insbesondere da sie bereits Buchstabenformen erwarten. Das HMAX-Modell hingegen ist (bisher) eine reine vorwärtsgerichtete Bottom-Up Architektur, die über keinerlei derartige a priori Informationen verfügt.

Diese Hypothese sollte daher in einer Folgestudie mit gesunden Probanden näher untersucht werden. Dabei sollten Stimuli verwendet werden, die dem Navon-Set sehr ähnlich sind, jedoch so wenige Top-Down Informationen (also Wissen um zu erwartende Stimuli etc.) wie möglich enthalten. So könnte man den Probanden beispielsweise kleine geometrische Elemente präsentieren, die in ihrer Gesamtheit entweder einen Buchstaben, ein anderes bekanntes oder aber gar kein Objekt ergeben. Zusätzlich müssten wieder die Inter-Element-Abstände systematisch variiert werden.

Weiterhin sollte das Modell mit weiteren Studien zur Simultanagnosie (beispielsweise Huberle et al., 2010; Huberle & Karnath, 2012; Thomas et al., 2012) aber auch solchen zur allgemeinen Gestaltwahrnehmung (beispielsweise Gregory, 1972; Grossberg & Mingolla, 1985; Littmann, Neumann, & Redouloux, 1996; Neumann & Mingolla, 2001) verglichen werden, um eventuelle Schwachpunkte oder noch nicht berücksichtigte Aspekte zu finden.

Gleichzeitig sollte natürlich auch das Modell sowie die verwendeten Stimuli erweitert und verbessert werden.

Während der Arbeit fielen einige Unstimmigkeiten des Stimulusmaterials auf, welche die Interpretation der Ergebnisse zum Teil erheblich erschwerten. Beispielsweise unterschied sich das Navon-A nicht nur in seiner Breite vom normalen A, sondern auch in seiner gesamten Größe von den übrigen Navon-Buchstaben. Zwar ergaben insbesondere die Studien zur Dickeninvarianz (vergleiche Abschnitt 3.1), dass dieses Modell bereits relativ gut mit gewissen Proportionsunterschieden umgehen kann. Für weitere Studien wäre einheitlicheres Material mit gleichen Proportionen allerdings vorteilhafter, um eine bessere Vergleichsgrundlage insbesondere im Hinblick auf normale Buchstaben zu haben.

Zudem sollten eventuell andere Buchstabentypen verwendet werden. Insbesondere die Studien zum Modellverhalten bei Präsentation mehrerer Buchstaben (siehe Abschnitt 3.3) ergaben, dass einige Buchstaben besser erkannt werden können als andere – was möglicherweise durch die relativ hohe Ähnlichkeit zueinander (beispielsweise B versus E oder H) bedingt sein könnte. Die Verwendung distinkterer Buchstaben könnte das Stimulusmaterial diesbezüglich vereinfachen.

In den vorliegenden Simulationen konnten zwar bereits viele wichtige Faktoren analysiert werden, auf einige Aspekte konnten im Rahmen dieser Arbeit jedoch nicht detailliert eingegangen werden. Raum für weiterführende Studien liefern beispielsweise die vielversprechenden Ergebnisse zur Präsentation mehrerer verschiedener Buchstaben. Zudem könnten neben den untersuchten Faktoren Größe, Position und Dicke auch noch weitere Bildeigenschaften eine entscheidende Rolle spielen. Besonders zu erwähnen ist dabei die Intensität der präsentierten Buchstaben. So mussten die Navon-Buchstaben stets im Training präsentiert werden, um anschließend auch im Testdurchgang erfolgreich erkannt werden zu können (siehe hierzu Abschnitt 4). Offensichtlich ist das Potential zur richtigen Klassifikation also vorhanden, durch die "Zerstückelung" der Buchstaben könnten die Antworten der notwendigen Features aber zu schwach sein, um vom Klassifikator – der ja ausschließlich auf geschlossene Buchstaben mit starken Featureantworten trainiert wurde – erkannt zu werden. Diesem Ansatz folgend wäre es also lohnenswert, auch die Generalisierbarkeit des Modells hinsichtlich variierender Buchstabenintensitäten zu untersuchen (beispielsweise mit einem Setting analog zur Größeninvarianz-Simulation), wodurch möglicherweise ein Training ohne Verwendung der Navon-Buchstaben erreicht werden könnte.

Nicht zuletzt sollten in weiterführenden Studien die bereits in Abschnitt 2.1 beschriebenen Erweiterungen integriert werden. Wie bereits in Abschnitt 1.1 erläutert ist es Patienten, bei denen eine Simultanagnosie

durch eine Läsion des PPC vorliegt, unter bestimmten Bedingungen durchaus möglich globale Eigenschaften zu verarbeiten. Die Vermutung ist also, dass diese Läsion dazu führt, dass die Aufmerksamkeit sich nicht auf globale Eigenschaften lenken lässt, wenn lokale Elemente vorliegen – der Aufmerksamkeitsfokus quasi "gelockt" ist. Interessanterweise lässt sich dieser Effekt auch umkehren (Thomas et al., 2012). Der PPC könnte also eine Art Schaltstelle sein, welche die Aufmerksamkeit aktiv auf verschiedene Auflösungen legen kann. Wäre diese gestört, könnte die Aufmerksamkeit, einmal auf einen bestimmten Fokus eingestellt, nicht mehr auf eine andere Auflösung gelegt werden. Trotzdem scheint es möglich zu sein, weiterhin nach anderen lokalen Eigenschaften zu suchen (Clavagnier, Fruhmann Berger, Klockgether, Moskau, & Karnath, 2006). Der grundsätzliche Mechanismus, um zwischen verschiedenen Lokationen bzw. Auflösungen umzuschalten, könnte die bereits in Abschnitt 2.1 erwähnte dynamische Selbstinhibition sein.

Um diesen Ansatz zu vertiefen und mit den in dieser Arbeit gewonnenen Erkenntnissen zu verbinden, schlagen wir basierend auf Chikkerur et al. (2010) eine Modellierung vor, bei der Lokationen, Auflösungen und Features in einem dynamischen Prozess miteinander in Konkurrenz stehen – wobei sich dieser Konkurrenzkampf wiederum durch Top-Down Prozesse (des PPC) beeinflussen lässt. In einem ersten Schritt findet eine Vorselektion möglicher Positionen statt. Diese könnte dem in dieser Arbeit verwendeten vorgeschalteten Aufmerksamkeitsfilter ähneln und vielversprechende Lokationen der Aufmerksamkeit für jede Filtergröße selektieren. Im Folgenden stehen die so ausgewählten Lokationen in Konkurrenz um die Aufmerksamkeit des Modelles. Jedes der wettstreitenden möglichen Patches besitzt drei Eigenschaften, die unterschiedlich gewichtet werden: Lokation, Auflösung und die Antworten der Features, welche zur Klassifikation dieses Patches ausschlaggebend sind. Die a priori Gewichtungen dieser Faktoren entsprechen dabei den Vorerfahrungen: 1) Das Gewicht der Lokation entspricht der Erfahrung, wo im Bild die gesuchten Eigenschaften typischerweise zu finden sind. 2) Das a priori Gewicht der verschiedenen Auflösungen repräsentiert die Erfahrung, welche Größe die gesuchten Eigenschaften wahrscheinlich besitzen. 3) Die Top-Down Gewichtung der einzelnen Features entspricht der Erfahrung, welche Features für die Klassifizierung eines bestimmten Objektes notwendig sind.

Während die a priori Gewichtungen die Vorerfahrungen modellieren, geht es bei der anschließenden Inhibition der drei Eigenschaften darum, den Fokus aktiv auf noch nicht beachtete Eigenschaften zu legen. Dies kommt insbesondere bei konkreten Aufgabenstellungen zum Tragen, bei denen ein gezieltes Abscannen des Bildes im Sinne konjunktiver Suchaufgaben erforderlich ist (bspw. "Suche den großen Buchstaben im Bild" oder "Suche ein A im Bild"). Hier wäre folgender Ansatz denkbar: Wird eine Lokation und Auflösung zu einem Zeitschritt gewählt und mit bestimmten Features eine Klassifikation durchgeführt, so inhibieren sich diese im nächsten

Zeitschritt selbst. Dadurch wird die Wahrscheinlichkeit erhöht, an einer anderen Position und/oder in einer anderen Auflösung nach einem neuen Objekt (in unserem Fall Buchstaben) zu suchen. Entsprechend der Aufgabenstellungen bzw. Erwartungshaltungen werden dabei die einzelnen Faktoren unterschiedlich stark gehemmt. Soll beispielsweise nach einem großen Buchstaben gesucht werden, so wird die aktuelle Lokation zwar stark inhibiert, um auch an anderen Stellen im Bild suchen zu können, die Auflösung hingegen wird nur geringfügig inhibiert, um auch weiterhin nach großen Objekten suchen zu können.

Natürlich müssen bei der Modellierung der Selbstinhibition noch weitere Faktoren berücksichtigt und insbesondere die biologische Plausibilität in weiteren neuropsychologischen Studien überprüft werden. So sollten Simultanagnosie-Patienten diesem Ansatz zufolge beispielsweise auch auf eine andere Auflösung "gelockt" werden können, wenn sie zuvor auf diese geprint wurden.

Gelingt es, mit diesem Modell die zahlreichen Facetten der Simultanagnosie zu modellieren, so würde dies wichtige Rückschlüsse auf noch immer nicht vollständig verstandenen Prozesse zur globalen Gestalt-Erkennung im Zusammenspiel mit dorsalem und ventralem Pfad erlauben.

---

## References

- Bálint, D. R. (1909). Seelenlähmung des "schauens", optische ataxie, räumliche störung der aufmerksamkeit. *European Neurology*, 25(1), 51–66.
- Chikkerur, S., Serre, T., Tan, C., & Poggio, T. (2010). What and where: A bayesian inference theory of attention. *Vision research*, 50(22), 2233–2247.
- Clavagnier, S., Fruhmann Berger, M., Klockgether, T., Moskau, S., & Karnath, H.-O. (2006). Restricted ocular exploration does not seem to explain simultanagnosia. *Neuropsychologia*, 44(12), 2330–2336.
- Gregory, R. (1972). Cognitive contours. *Nature; Nature*.
- Grossberg, S., & Mingolla, E. (1985). Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentations. *Perception & psychophysics*, 38(2), 141–171.
- Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3), 574–591.
- Huberle, E., Driver, J., & Karnath, H.-O. (2010). Retinal versus physical stimulus size as determinants of visual perception in simultanagnosia. *Neuropsychologia*, 48(6), 1677–1682.
- Huberle, E., & Karnath, H.-O. (2006). Global shape recognition is modulated by the spatial distance of local elementsevidence from simultanagnosia. *Neuropsychologia*, 44(6), 905–911.
- Huberle, E., & Karnath, H.-O. (2012). The role of temporo-parietal junction (tpj) in global gestalt perception. *Brain Structure and Function*, 217(3), 735–746.
- Itti, L., & Koch, C. (2001). Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3), 194–203.
- Kanizsa, G. (1955). Margini quasi-percettivi in campi con stimolazione omogenea. *Rivista di psicologia*, 49(1), 7–30.
- Karnath, H.-O., Ferber, S., Rorden, C., & Driver, J. (2000). The fate of global information in dorsal simultanagnosia. *Neurocase*, 6(4), 295–306.
- Koch, C., & Ullman, S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence* (pp. 115–141). Springer.
- Littmann, E., Neumann, H., & Redouloux, L. (1996). Extraction of illusory contours by perceptual grouping. In *Mustererkennung 1996* (pp. 243–251). Springer.
- Luria, A. R. (1959). Disorders of a "simultaneous perception" in a case of bilateral occipitoparietal brain injury. *Brain*, 82(3), 437–449.
- Mutch, J., & Lowe, D. G. (2006). Multiclass object recognition with sparse, localized features. In *Computer vision and pattern recognition, 2006 iee computer society conference on* (Vol. 1, pp. 11–18).

## References

---

- Navon, D. (1977). Forest before trees: The precedence of global features in visual perception. *Cognitive psychology*, *9*(3), 353–383.
- Neumann, H., & Mingolla, E. (2001). 12 computational neural models of spatial integration in perceptual grouping. *Advances in Psychology*, *130*, 353–400.
- Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience*, *2*(11), 1019–1025.
- Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., & Poggio, T. (2005). *A theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex* (Tech. Rep.). DTIC Document.
- Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., & Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *29*(3), 411–426.
- Thomas, C., Kveraga, K., Huberle, E., Karnath, H.-O., & Bar, M. (2012). Enabling global processing in simultanagnosia by psychophysical biasing of visual pathways. *Brain*, *135*(5), 1578–1585.
- Wolpert, I. (1924). Die SimultanagnosieStörung der Gesamtauffassung. *Zeitschrift für die gesamte Neurologie und Psychiatrie*, *93*(1), 397–415.