

**Studienarbeit**  
**Sommersemester 2001**

**Parameterbestimmung für die  
Dosisregelung von Durchleuchtungssystemen**

Sami Celik

Matrikelnr. 209820

Betreuer

Dipl.-Inform. Jörg Dahmen

Dipl.-Inform. Daniel Keysers

**RWTH Aachen**  
**Lehrstuhl für Informatik VI**  
**Prof. Dr.-Ing. H. Ney**

**In Zusammenarbeit mit den  
Philips GmbH Forschungslaboratorien, Aachen**

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Problemstellung</b>	<b>5</b>
2.1	Prinzip eines Durchleuchtungssystems . . . . .	5
2.2	Beschreibung des Problems . . . . .	5
2.3	Bestimmung der Referenzwerte mittels Segmentierung . . . . .	7
<b>3</b>	<b>Klassifikation und Regression</b>	<b>8</b>
3.1	Grundlagen . . . . .	8
3.2	Das allgemeine Regressionsproblem . . . . .	9
3.3	Methode der kleinsten Quadrate . . . . .	10
3.4	Merkmalsanalyse . . . . .	11
3.5	Formulierung des Parameterbestimmungsproblems . . . . .	12
<b>4</b>	<b>Methoden</b>	<b>12</b>
4.1	Die 60%-Methode . . . . .	13
4.2	Nearest-Neighbour . . . . .	15
4.3	Kernel-Densities . . . . .	16
4.4	Neuronale Netze . . . . .	17
4.4.1	Das Multilayer-Perceptron . . . . .	18
4.4.2	Neuronales Netz für die Parameterbestimmung . . . . .	21
<b>5</b>	<b>Implementierung</b>	<b>23</b>
<b>6</b>	<b>Ergebnisse und Diskussion</b>	<b>25</b>
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>31</b>

## Zusammenfassung

*In der vorliegenden Arbeit wird ein Regressionsproblem aus dem Bereich der Röntgendiagnose untersucht. Bei einer statischen Röntgenaufnahme ist die Einstellung der Strahlendosis im Normalfall einmalig. Die Durchleuchtung ist eine dynamische Variante der Röntgendiagnostik bei der die Dosis anhand eines Parameters ständig zur Laufzeit geregelt werden muss. Die Bestimmung des Parameters wurde in dem betrachteten Durchleuchtungssystem bisher unter Verwendung der mittleren Strahlungsdichte des zentralen Bildteils durchgeführt, was jedoch keine optimale Anpassung gewährleistet. Abstandsklassifikatoren und neuronale Netze die zur Bestimmung des Parameters eingesetzt werden können, versprechen vergleichsweise bessere Anpassung und erfüllen die Kriterien an die Laufzeit.*

## 1 Einleitung

Das Interesse, den menschlichen Körper zu untersuchen, die biologischen Abläufe in ihm zu verstehen, Krankheiten zu diagnostizieren und zu heilen sind und waren schon immer die Charakteristiken der Medizin. Mit der Entwicklung der Technik sind viele Untersuchungen erleichtert oder sogar erst ermöglicht worden. Dazu zählt insbesondere das sogenannte Röntgenbild. Eine Röntgenuntersuchung ist die bildliche Darstellung unterschiedlicher Körperregionen und -gewebe mit Hilfe von Röntgenstrahlen. Die Röntgenstrahlen wurden 1895 von dem deutschen Physiker Wilhelm Konrad Röntgen entdeckt. Heute sind Röntgensysteme für medizinische Untersuchungen nicht mehr aus dem klinischen Alltag wegzudenken.

Die bekannteste Form einer Röntgenuntersuchung ist die statische Röntgenaufnahme. Hierbei wird das Objekt fest positioniert und die Röntgenaufnahme durchgeführt. In der Regel wird bei dieser Form nur hartes Gewebe aufgenommen (mit Ausnahme z.B. der Mammographie). Dabei ergeben sich durch die unterschiedliche Strahlenabsorption im Gewebe natürliche Kontraste. Weiches Gewebe wie Fett, Muskeln und Haut absorbiert wenig Strahlung und schwärzt den Röntgenfilm. Hartes Gewebe wie Knochen absorbiert dagegen viel Strahlung und hinterlässt dadurch einen weißen Schatten auf dem Röntgenbild. In dieser Arbeit geht es um die Röntgendurchleuchtung, welches ein weiteres Verfahren der Röntgendiagnostik darstellt. Im Unterschied zum statischen Röntgen ist die Durchleuchtung eine dynamische Variante dieser Untersuchungsform. Ein entscheidender Vorteil ist, dass zeitlich ablaufende Vorgänge im Körper sichtbar gemacht werden können. Dabei wird die aus dem Körper austretende Röntgenstrahlung, also die nicht absorbierte Strahlung, mit Hilfe einer Bildverstärker-Fernsehkette auf einem Fernsehmonitor als Bild dargestellt. Der Arzt kann auf diese Weise funktionelle Abläufe im Körper kontinuierlich auf dem Monitor beobachten.

Damit mittels der Röntgenstrahlen auch weiches Gewebe sichtbar gemacht werden kann, werden häufig Röntgen-Kontrastmittel eingesetzt, wodurch künstliche Kontraste auf dem Röntgenbild bzw. auf dem Fernsehmonitor entstehen. Kontrastmittel sind nötig, um die Dichte bestimmter Strukturen zu erhöhen und diese dadurch deutlich sichtbar zu machen. Beispielsweise lassen sich Hohlräume wie der Verdauungstrakt, aber auch Gefäße deutlich hervorheben und dadurch besser untersuchen. Auf dem Röntgenbild bzw. dem Fernsehmonitor wird das Ergebnis als Schwarzweiß-Kontrast dargestellt. So ist z.B. ein mit Röntgen-Kontrastmittel gefüllter Magen im Unterschied zur restlichen Umgebung auf dem Röntgenbild bzw. Monitor weiß. Der weiße Kontrast kommt dadurch zustande, dass das Röntgen-Kontrastmittel die Röntgenstrahlung absorbiert.

In der Praxis kann der Arzt eine Röntgendurchleuchtung veranlassen, wenn ein Patient beispielsweise über Schluckstörungen klagt. Der Patient bekommt in diesem Fall ein Röntgen-

Kontrastmittel zum Trinken. Der ganze Trinkvorgang kann nun mittels Durchleuchtung auf einem Monitor verfolgt werden. Auf diese Weise ist es dem Arzt möglich, beispielsweise Bewegungsstörungen der Speiseröhre festzustellen. Die Röntgendurchleuchtung kommt auch bei Operationen zum Einsatz. Dabei kann der Arzt beurteilen, ob ein Herzschrittmacher richtig positioniert oder ob eine Schraube optimal im Knochen angebracht ist [1].

Ein großer Nachteil einer Röntgenuntersuchung ist die Strahlenbelastung des Patienten. Die Strahlendosis bei einer einfachen Röntgenuntersuchung ist im Vergleich zur natürlichen Strahlung, der der Körper das ganze Jahr über ausgesetzt ist, gering. Doch jede zusätzliche Röntgenaufnahme erhöht die Strahlenbelastung des Körpers. Es ist zu bedenken, dass selbst bei vielen Röntgenaufnahmen innerhalb kurzer Zeit das Gesundheitsrisiko einer nicht erkannten Krankheit meistens wesentlich größer ist. Bei der Durchleuchtung kann wegen der dauerhaften Strahlenexposition nur mit geringer Intensität der Strahlung gearbeitet werden. Trotz moderner Technik ist die Durchleuchtung mit einer relativ hohen Strahlenbelastung verbunden, sodass versucht wird, die Durchleuchtungszeit möglichst kurz zu halten [1].

Vor allem aus Gründen der Strahlenbelastung sind hohe Anforderungen an ein Röntgedurchleuchtungssystem gestellt. Arbeitet das Durchleuchtungssystem langsam und fehleranfällig, so wird die Untersuchung länger dauern, was zu Folge hat, dass der Patient länger bestrahlt wird. Um z.B. den Bewegungsablauf des Trinkens darstellen zu können, werden viele Röntgenbilder hintereinander erzeugt, wodurch eine kontinuierliche Verfolgung ermöglicht wird. An dieser Stelle sei bemerkt, dass hier vollautomatische Durchleuchtungssysteme betrachtet werden, d.h. die Strahlendosis wird anhand bestimmter Parameter zu jeder Aufnahme automatisch geregelt. Dabei sollte die Dosis immer so eingestellt sein, dass das Bild auf dem Monitor in möglichst gutem Kontrast dargestellt wird. In bestimmten Übergangsregionen, beispielsweise vom Kopf in den Halsbereich oder vom Hals in den Oberkörperbereich, sind die Bedingungen für das System erschwert, da die Absorptionsverhältnisse ganz unterschiedlich sind. Wird in solch einer Phase der Regelparameter vom System nicht korrekt bestimmt, wird folglich die Bestrahlungsdosis falsch eingestellt und schließlich das Bild auf dem Monitor zu dunkel oder zu hell. Der Arzt muss diesen Vorgang im Halsbereich eventuell mehrmals wiederholen, wodurch die Bestrahlung des Patienten länger dauert. Das Röntgensystem soll solche Situationen schnell erkennen und die Dosis möglichst gut einstellen.

In dieser Arbeit geht es um die Parameterbestimmung zur Dosisregelung von Röntgensystemen. Die Parameterbestimmung kann mittels Algorithmen der Bildverarbeitung durchgeführt werden. Bisher wurde dies in dem betrachteten System durch eine sogenannte 60%-Methode durchgeführt. Hierbei wird der Parameter über die mittlere Helligkeit der mittleren Teils des Röntgenbildes bestimmt. Die Methode scheint jedoch nicht generalisierbar zu sein, denn sie führt in zahlreichen Situationen zu Fehlern [15]. Aus diesem Grund werden hier weitere Methoden aus der Bildverarbeitung vorgestellt, die für die Parameterbestimmung eingesetzt werden können und auch bessere Ergebnisse zeigen. Zunächst wird im nächsten Abschnitt die Problemstellung detaillierter beschrieben. Es zeigt sich, dass das Problem der Parameterbestimmung einer allgemeinen Klasse von Problemen zugeordnet werden kann, nämlich der Regression (Abschnitt 3). Typische Probleme der Bildverarbeitung sind die verwandten Klassifikationsprobleme. In Abschnitt 4 werden die verschiedene Methoden vorgestellt und es wird gezeigt, wie diese dem vorliegenden Problem angepasst werden können. Schließlich werden in Abschnitt 5 die Ergebnisse präsentiert, die im Rahmen dieser Studienarbeit ermittelt wurden.

## 2 Problemstellung

Im Folgenden wird das Prinzip eines Röntgendurchleuchtungssystems erläutert. Danach wird auf die Problematik der bisher verwendeten sogenannten 60%-Methode eingegangen. Weiter werden Vor- und Nachteile von Segmentierungsalgorithmen im Kontext der Parameterbestimmung diskutiert.

### 2.1 Prinzip eines Durchleuchtungssystems

Wie schon in der Einleitung beschrieben wurde, kann durch die dauerhafte Strahlenexposition während der Durchleuchtung nur mit geringer Intensität der Strahlung gearbeitet werden. Die Auflösung und Detailerkennbarkeit ist viel geringer als bei der Einzelaufnahme. Damit die Strahlenausnutzung optimal geregelt werden kann, sind heutige Durchleuchtungssysteme rückgekoppelt. Abb.1 zeigt das Prinzip eines Durchleuchtungssystems [2]. Die Röntgenstrahlen durchleuchten das Objekt und treffen auf den fluoreszierenden Eingangsschirm des Röntgenbildverstärkers. Durch Hochspannung werden die dabei ausgelösten Elektronen auf den Ausgangsschirm des Röntgenbildverstärkers beschleunigt und fokussiert. Diese werden über ein optisches System auf eine Fernsehkamera abgebildet und schließlich auf dem Monitor angezeigt. Die Helligkeit des Durchleuchtungsbildes wird durch eine Steuerelektronik ständig gemessen und über einen Ist-Sollwert-Vergleich werden Strom und Spannung, welche die Röntgendosis steuern, ständig nachgeregelt, sodass das Bild immer optimal ausgeleuchtet ist. Das System regelt automatisch die Dosis nach, wenn Abweichungen bei dem Ist-Sollwert-Vergleich auftreten sollten. Während der Durchleuchtung wird das Bild ständig auf dem Monitor angezeigt und ermöglicht somit die Beurteilung dynamischer Vorgänge.

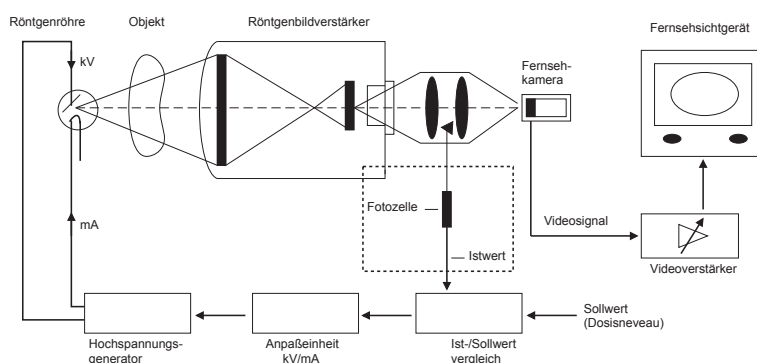


Abbildung 1: Prinzip eines Durchleuchtungssystems. Diese Arbeit beschäftigt sich mit der Istwert-Bestimmung (markierter Bereich). [2]

### 2.2 Beschreibung des Problems

Im Mittelpunkt dieser Studienarbeit steht nur eine bestimmte Komponente des Durchleuchtungssystems, nämlich die Bestimmung der Helligkeit des Ausgangsschirmes. In den derzeitigen Röntgensystemen erfolgt die Regelung der Dosis über die Beobachtung der mittleren Helligkeit des mittleren Teiles des Ausgangsschirmes des Röntgenbildverstärkers (60%-Methode). Dieser Vorgang entspricht etwa der Spotmessung in der Fotografie. Die Objektmessung ist die am weitesten verbreitete Methode der Belichtungsmessung in der Fotografie [3]. Das aufzunehmende Objekt wird vom Kamerastandort aus angepeilt und durch Messung des reflektierenden Lichts

die Belichtung bestimmt. Dabei wird i.A. zwischen einer Integralmessung und einer Spotmessung unterschieden. Bei der Integralmessung wird die mittlere Helligkeit des auf dem Foto landenden Lichtes gemessen. Bei der Spotmessung wird das Objekt mittels eines Suchers auf einen zentralen Bereich beschränkt, wodurch nur noch ein kleiner Anteil des reflektierten Lichts gemessen wird. Die von den derzeitigen Röntgensystemen verwendete Methode zur Belichtungsmessung wird im Folgenden anhand eines einfacheren Modells erläutert (Abb. 2).

Das Röntgensystem ermittelt zunächst ein digitales Bild zum aktuell durchleuchteten Objekt. Aus dem Bild werden Informationen extrahiert, mit denen es möglich wird, die Strahlendosis so einzustellen, dass das Objekt bei der nächsten Aufnahme möglichst gut ausgeleuchtet ist. Die Extraktion der Informationen aus einem Bild ist ein typischer Vorgang für die Bildverarbeitung und Mustererkennung. Dieser hier ermittelte Parameter, im Folgenden auch Grauwertparameter oder kurz Grauwert genannt, ist wiederum die Eingabe zur Dosisregelung für die nächste Aufnahme. Die Einstellung der Dosis selber wird in dieser Arbeit nicht weiter betrachtet.

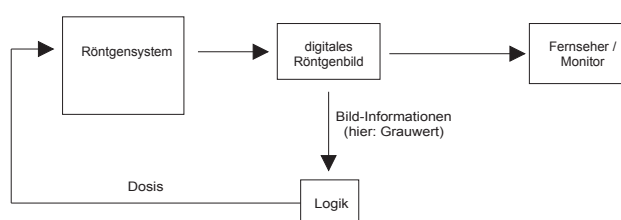


Abbildung 2: Ablauf bei der Röntgendurchleuchtung

Wie schon oben beschrieben worden ist, wird der Grauwertparameter bei den derzeit verwendeten Röntgensystemen über die mittlere Helligkeit der zentralen 60% der Bildelemente (Pixel) des Ausgangsschirmes bestimmt. Diese Methode wird im folgenden auch als 60%-Methode bezeichnet. Diese Technik führt in zahlreichen Situationen zu Fehlbelichtungen aufgrund von in das Messfeld einfallender Direktstrahlung [15]. Die Ursache dafür ist, dass aus dem aktuellen Bild ein falscher Grauwert bestimmt wird, was zur Folge hat, dass die Dosis zu hoch oder zu niedrig eingestellt wird. Die nächste Aufnahme wird folglich mit einer falschen Strahlendosis durchgeführt, wodurch es zu einer Fehlbelichtung kommt. Es ist an dieser Stelle zu bemerken, dass das Objekt in der Folgeaufnahme leicht verschoben ist. Die Verteilung der Grauwerte in dem Folgebild sind aber nur leicht verändert. Ist der ermittelte Grauwert für die aktuelle Aufnahme nahezu optimal, so wird die Folgeaufnahme auch gut genug ausgeleuchtet werden, denn kleinere Abweichungen werden hierbei toleriert. Abb. 3 zeigt Beispielder von Durchleuchtungsaufnahmen. Zu Fehlbelichtungen kann es in Übergangsbereichen wie z.B. beim Übergang vom Kopf- in den Halsbereich kommen, da sich das Objekt vollständig verändert. Eine Fehlbelichtung kann im ungünstigen Fall dazu führen, dass der Fehler verstärkt wird, weil die Messmethode den Grauwert einer fehlbelichteten Aufnahme schätzen muss. Dies kann dazu führen, dass eine komplette Sequenz von Aufnahmen fehlbelichtet wird. Das Interesse besteht nun darin, dass die Messmethode, vor allem bei Abweichungen, selbständig die richtige Einstellung findet, d.h. auch bei einer fehlbelichteten Aufnahme sollte der Grauwert möglichst gut geschätzt werden. Der hier verfolgte Ansatz besteht dabei darin, dass das System die Verteilung der Grauwerte im Bild, die typisch ist für bestimmte durchleuchtete Objekte, analysiert, um die Grauwertbestimmung durchzuführen.

Ziel dieser Arbeit ist es, weitere Methoden zu untersuchen, die zur Messsignalerzeugung verwendet werden können. Um den Zusammenhang zwischen Grauwertbestimmung und Bild-

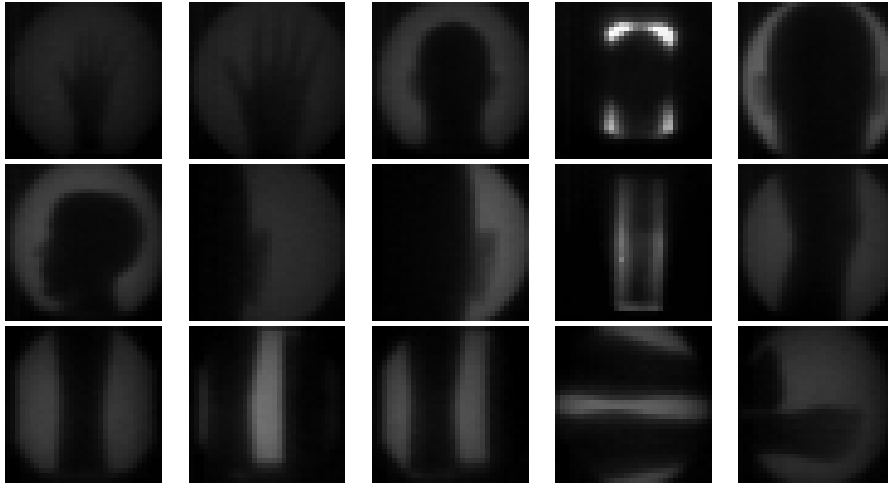


Abbildung 3: Beipielbilder von Durchleuchtungsaufnahmen (kleine Datenbank)

verarbeitung zu verdeutlichen, wird im folgenden die in Abb. 4 gezeigte Problemstellung betrachtet. Das Röntgensystem erzeugt aus den Informationen am Ausgangsschirm ein digitales Röntgenbild. Die Bilder haben hier eine Größe von  $32 \times 32$ -Pixel. Aus einem Bild werden mittels Werkzeugen der Bildverarbeitung spezifische Bildmerkmale bestimmt (Extraktion geeigneter Merkmale), d.h. eine stark reduzierte Darstellung des Bildes wird erreicht. Bei der 60%-Methode entspricht die Merkmalsextraktion dem Vorgang, dass man die Pixel des Bildes bestimmt die den 60%-Bereich ausmachen. Die Methoden sind i.A. Algorithmen, die mittels der reduzierten Bilddaten einen Grauwert ermitteln. Bei der 60%-Methode wird der Mittelwert der Grauwerte berechnet, die im vorigen Schritt bestimmt worden sind. Die Ausgabe des Algorithmus ist schließlich ein geschätzter Grauwert, der dem Regelsystem geliefert wird.

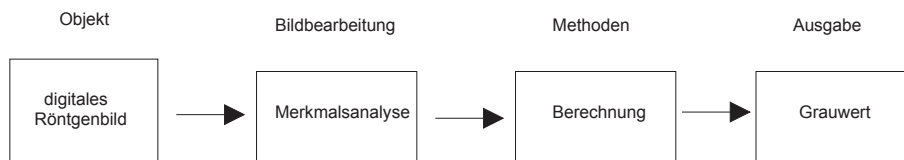


Abbildung 4: Die Schritte bei der Bestimmung des Grauwertes

### 2.3 Bestimmung der Referenzwerte mittels Segmentierung

Für diese Arbeit wurde eine Datenbank mit 82 digitalen Röntgenbildern zur Verfügung gestellt und eine weitere Datenbank mit 608 Bildern wurde im Verlauf noch ergänzt [15]. Für die Bilder der kleinen Datenbank wurden die (optimalen) Grauwerte von einem Experten bestimmt. Die optimalen Werte dienen als Referenz zu den Ergebnissen der untersuchten Methoden. Man spricht hier auch von gelabelten Daten (Abschnitt 3). Für die große Datenbank wurden die optimalen Werte mit einem Segmentierungsalgorithmus bestimmt. Segmentierer versuchen die in einem digitalen Bild (hier Röntgenbilder) vorkommenden Objekte (z.B. Knochen, Weichteile oder Organe) voneinander zu trennen. Im einfachsten Fall ist nur ein Objekt vorhanden, sodass es ausreichend ist, das Bild in 2 Regionen aufzuteilen. Dabei wird jedes Pixel entweder zum Objekt oder zum Hintergrund zugeordnet. In [2] wird Segmentierung wie folgt definiert. „Die hervorgehobene Darstellung von inhaltlich zusammenhängenden Regionen durch Zusammenfassung benachbarter Bildpunkte, die einem bestimmten Homogenitätskriterium genügen,

bezeichnet man als *Segmentierung*. Die anschließende Zuordnung dieser Regionen zu bestimmten Klassen, also zu einer inhaltlichen Bedeutung (Interpretation) entspricht der Klassifikation und Mustererkennung“.

Die Referenzwerte der Bilder der großen Datenbank sind mit einem Segmentierungsalgorithmus bestimmt worden. Bei der kleinen Datenbank wurden die Referenzwerte von einem Experten bestimmt. Wie schon beschrieben, ist der Grauwert eines Röntgenbildes zu bestimmen, welcher das Objekt (oder auch mehrere Objekte), in möglichst gutem Kontrast zum Hintergrund oder zu einem anderen auf dem Bild vorhandenen Objekt darstellt. Segmentierer sind in diesem Zusammenhang geeignete Verfahren zur Bestimmung der gesuchten Grauwerte, da sie die Berechnung des Parameters auf den Mittelwert der Objekt-Grauwerte zurückführen, die alleine interessant sind. Die Beleuchtung des Hintergrundes ist i.A. nicht interessant. Der Segmentierungsalgorithmus versucht, die Anzahl der Regionen eines Bildes zu bestimmen. Ist nur ein Objekt auf dem Bild vorhanden, dann kann der Hintergrund i.A. einfach von den Objekten getrennt werden, da der Grauwert des Hintergrundes dem Benutzer bekannt ist. Schwieriger wird es, wenn zwei oder mehrere Objekte zu unterscheiden sind. In diesem Fall ist die Berechnungsdauer des Segmentierers wesentlich länger. Es kann aber auch passieren, dass in schwierigeren Fällen, die Objekte durch den Segmentierer nicht getrennt werden können, d.h. es gibt Fälle bei denen der Algorithmus instabil ist. Sind die Regionen bestimmt, dann können diese analysiert werden und es kann schließlich ein Grauwert bestimmt werden.

Aus Gründen der hohen Laufzeit und Instabilität ist ein Segmentierer, wie er hier betrachtet wurde, nicht für die hier verwendeten Röntgensysteme geeignet. Nach Aufgabenstellung soll die verwendete Methode eine Laufzeit von maximal 20ms haben [15]. Aus diesem Grund sind andere Methoden gesucht, die das Kriterium der Laufzeit erfüllen und auch gute Ergebnisse bei der Ermittlung der Grauwerte zeigen. In Abschnitt 4 werden Methoden vorgestellt, die die geforderten Laufzeiten erfüllen und dabei auch gute Ergebnisse bei der Grauwertbestimmung zeigen. Diese sind Abstandsklassifikatoren und neuronale Netze. Zunächst werden im nächsten Abschnitt allgemeine Grundlagen der Klassifikation und Regression erläutert.

## 3 Klassifikation und Regression

### 3.1 Grundlagen

Klassifikation bedeutet im Sinne der Mustererkennung, die Zuordnung eines Testmusters in eine von mehreren möglichen Zielklassen. Allgemein wird eine Stichprobe von Mustern eingeteilt in eine Trainings- und Testmenge. Die Muster werden in 2-Tupel Form dargestellt

$$(x_i, k_i) \in \mathbb{R}^d \times \mathbb{K} \quad \text{mit } \mathbb{K} = \{1, \dots, K\}, \quad (1)$$

wobei  $k_i$  die richtige Klasse des Musters  $i$  ist. Geeignete Merkmale aus Muster  $i$  bilden den Merkmalsvektor  $x_i \in \mathbb{R}^d$ . Die Komponenten von  $x_i$  quantifizieren die Eigenschaften des Musters  $i$  im Merkmalsraum  $\mathbb{R}^d$ . Der Übergang vom Muster zum Merkmalsvektor wird Merkmalsanalyse genannt. Die in dieser Arbeit vorgestellten Verfahren arbeiten nach dem Prinzip des „überwachten Lernens“, d.h. die Zielklassen  $k_i$  sind im Training bekannt. Mit einer geeigneten Merkmalsextraktion kann das Klassifikationsproblem folgendermaßen formuliert werden. Generierung einer spezifischen Entscheidungsregel

$$r : \mathbb{R}^d \mapsto \{1, \dots, K\} \quad (2)$$

$$r(x_i) = \hat{k}_i \quad x_i \in \mathbb{R}^d, \quad \hat{k}_i \in \{1, \dots, K\} \quad (3)$$

mittels Trainingsdaten. Der Klassifikator  $r$  ordnet einem beliebigen Merkmalsvektor  $x_i$  eine Klasse  $\hat{k}_i$  zu. Wir nehmen an, dass die richtige Klasse  $k_i$  zu einem Muster  $i$  bekannt sei. Dann



nennt man den Fall  $k_i \neq \hat{k}_i$  einen Fehler. Die Fehlerrate eines Klassifikators ist definiert als der Quotient zwischen Summe der Gesamtfehler und der Anzahl der Testmuster.

Bei der Parameterbestimmung zur Dosisregelung ist ein Regressionsproblem zu lösen. Regression ist die allgemeinere Form der Klassifikation und ist ein statistisches Verfahren zur Untersuchung der Beziehungen zwischen Variablen. Man möchte feststellen, ob z.B. eine Variable als Funktion von anderen Variablen ausgedrückt werden kann oder Werte einer Variablen durch die von anderen Variablen vorhersagen kann [7]. Für die Darstellung der Muster ergibt sich im allgemeinen Fall

$$(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}^{d_L}$$

Im Vergleich zu (1), wo die Lösung eindimensional und diskret ist, quantifizieren die Komponenten von  $y_i$  im allgemeineren Fall einen kontinuierlichen Lösungsraum der Dimension  $d_L$ . Für die Abbildungsvorschrift (2) und (3) ergibt sich die allgemeine Form mit

$$s : \mathbb{R}^d \mapsto \mathbb{R}^{d_L} \tag{4}$$

$$s(x_i) = \hat{y}_i \quad x_i \in \mathbb{R}^d, \hat{y}_i \in \mathbb{R}^{d_L} \tag{5}$$

Unter der Annahme, dass  $y_i \in \mathbb{R}^{d_L}$  das richtige Ergebnis ist, muß für den Fall  $\hat{y}_i \neq y_i$  eine geeignete Fehlerfunktion benutzt (oder definiert) werden. Hier wird häufig die euklidische Distanzfunktion benutzt. Diese ist gegeben durch die Abbildung

$$d : \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \mapsto \mathbb{R}$$

$$d(a, b) = \sum_{i=1}^{d_L} (a_i - b_i)^2 \quad a, b \in \mathbb{R}^{d_L} \tag{6}$$

### 3.2 Das allgemeine Regressionsproblem

Mit den gegebenen Trainingsdaten muß also eine Funktion gefunden werden, die die in (4) geforderte Abbildung realisiert und dabei möglichst geringe Abweichungen vom Zielwert erzeugt. Dazu wird im folgenden die klassische Formulierung des Regressionsproblems betrachtet [4]. Seien im folgenden  $d = 1$  und  $d_L = 1$ . Ein Vektor  $w_0$  aus Parametern einer Abbildungsfunktion  $g(x, w_0)$  ist gesucht. Um die Parameter zu approximieren, wird in jedem beobachteten Punkt  $x_i \in \mathbb{R}$  (das sind die Muster) ein Fehler  $\varepsilon_i \in \mathbb{R}$  definiert

$$y_i = g(x_i, w_0) + \varepsilon_i \tag{7}$$

wobei der Fehler als unabhängig von  $x$  angenommen wird und einer Dichteverteilung  $p_\varepsilon(\varepsilon)$  entspricht. Aufgrund der beobachteten Daten (Muster)  $Z = \{(x_i, y_i) \mid i = 1, \dots, n\}$  ergibt sich die Log-Likelihood-Funktion durch

$$\ln P(Z | w) = \sum_{i=1}^n \ln p_\varepsilon(y_i - g(x_i, w))$$

Unter der Annahme, dass der Fehler normalverteilt ist und eine mittleren Varianz  $\sigma^2$  aufweist, ergibt sich für die Log-Likelihood-Funktion

$$\ln P(Z | w) = -\frac{1}{2\sigma^2} \left[ \sum_{i=1}^n (y_i - g(x_i, w))^2 \right] - n \ln(\sqrt{2\pi}\sigma)$$

Maximieren der Log-Likelihood-Funktion ist daher äquivalent zum Minimieren der Funktion

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - g(x_i, w))^2 \quad (8)$$

Das Minimierungsproblem (8) ist die sogenannte empirische Risikominimierung, d.h.  $g$  wird dahingehend optimiert, dass eine möglichst geringe Zahl von Trainingsabweichungen auftritt. Durch die Minimierung der Abweichung im Training wird eine gute Leistung bei der Erkennung unbekannter Testdaten erhofft. Problematisch dabei ist, wie „repräsentativ“ die Menge der Trainingsdaten für die Menge alle auftretenden Daten ist. Insbesondere für kleine Mengen von Trainingsdaten besteht die Gefahr, dass der Klassifikator eine schlechte Fähigkeit zu Generalisierung hat. Abb. 5 (a) zeigt qualitativ eine nichtlineare Regressionskurve.

### 3.3 Methode der kleinsten Quadrate

Es wird im Folgenden als Spezialfall die lineare Regression betrachtet. Diese soll als Beispiel dienen, um zu zeigen, dass dieser einfache Fall, mittels der Methode der kleinsten Quadrate, einfach zu lösen ist. Für den linearen Fall kann Gleichung (7) wie folgt formuliert werden.

$$y_i = a + b \cdot x_i + \varepsilon_i \quad i = 1, 2, \dots, n \quad (9)$$

Die gegebenen Trainingsdaten  $(x_1, y_1), \dots, (x_n, y_n)$  erzeugen eine Punktwolke im zweidimensionalen Raum (Abb. 5 (b)). Gesucht ist die beste Interpolationsgerade, d.h. die Gerade, für welche die Abweichungen der Interpolationswerte  $\hat{y}_i = a + b \cdot x_i$  von den optimalen Werten  $y_i$  minimal werden. Abb. 5 (b) zeigt qualitativ eine Regressionsgerade.

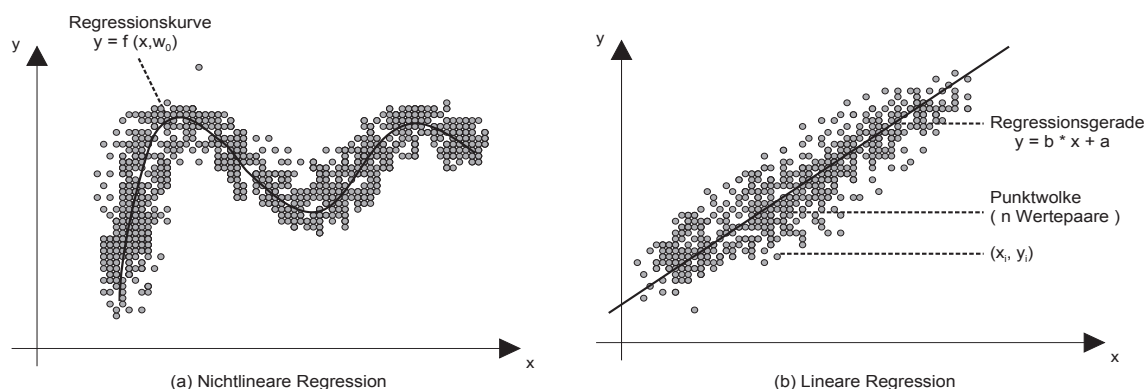


Abbildung 5: Nichtlineare und lineare Regression

Mit der Methode der kleinsten Quadrate (least squares method) kann der Ansatz (9) gelöst werden [5] [12]. Durch Umformulierung ergibt sich für den Fehler

$$\varepsilon_i = y_i - (a + b \cdot x_i) \quad i = 1, 2, \dots, n$$

Die Parameter  $a$  und  $b$  werden so geschätzt, dass

$$\sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n \underbrace{(y_i - a - bx_i)^2}_{g(a,b)} \rightarrow \text{Min}$$

(Minimierung über  $a$  und  $b$ ), d.h. die Summe der Abweichungsquadrate wird minimiert. Eine notwendige Bedingung für die Existenz eines Extremwertes ist das Verschwinden der beiden partiellen Ableitungen nach  $a$  und  $b$  mit

$$\frac{\partial g(a, b)}{\partial b} = 0 \quad \text{und} \quad \frac{\partial g(a, b)}{\partial a} = 0$$

Nach Ausführen der partiellen Differentiation, Einsetzen in die obigen Gleichungen und Ausführen der Summation erhält man ein Gleichungssystem für  $a$  und  $b$ , die Normalgleichungen des Regressionsproblems mit den Lösungen

$$b_{\text{LS}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$a_{\text{LS}} = \bar{y} - b_{\text{LS}} \cdot \bar{x}$$

mit

$$\bar{x} = \frac{1}{n} \sum x_i \quad \text{und} \quad \bar{y} = \frac{1}{n} \sum y_i$$

Damit ergibt sich die geschätzte Regressionsgerade

$$y_{\text{LS}}(x) = a_{\text{LS}} + b_{\text{LS}} \cdot x$$

Der allgemeine Fall der Regression ist jedoch i.A. nicht einfach zu lösen. Hierzu sind mächtigere Methoden notwendig. Ausserdem müssen die Methoden dem jeweiligen Problem angepasst werden. Beispielsweise sind neuronale Netze ein geeigneter Ansatz zur Behandlung allgemeiner Regressionsprobleme, wie auch im vorliegenden Fall der Grauwertbestimmung. In Abschnitt 3.5 wird versucht das vorliegende Problem formal zu beschreiben.

### 3.4 Merkmalsanalyse

In diesem Abschnitt wird auf die Merkmalsanalyse eingegangen, die in dieser Arbeit verwendet wird. Es gibt verschiedene Methoden, um Merkmale eines Bildes zu beschreiben. Die hier gewählte Methode entspricht der Aufgabenstellung nach [15]. Die Bilder haben eine Größe von  $32 \times 32$ -Pixel. Der Grauwertbereich wird mit  $G$  (hier  $G = 256$ ) angegeben, d.h. es gibt Graustufen im Bereich von 0 bis 255. Ein Bild wird zunächst für ein vorgegebenes  $n \in \{1, \dots, 32\}$  in  $n^2$  symmetrische Felder aufgeteilt. Der mittlere Grauwert jedes Feldes bildet das Merkmal zu diesem Feld. Durch dieses Vorgehen ergeben sich  $n^2$  Merkmale die einen Merkmalsvektor der Form

$$x_i = (x_{i1}, \dots, x_{in^2}) \quad \text{mit} \quad x_{ij} \in \mathbb{R},$$

bilden. Die Dimension des Merkmalraumes entspricht genau der Anzahl der Felder in die ein Bild eingeteilt wird, d. h.  $d = n^2$ . Abb. 6 skizziert die Merkmalsextraktion am Beispiel für  $n = 3$ . Die Grenzen der eingeteilten Felder sind durch die  $t_i$  gekennzeichnet. Beispielsweise lautet die Berechnung für Merkmal  $x_{i5}$  (bei  $n = 3$ ,  $a_{jk}$  ist der Grauwert eines Pixels).

$$x_{i5} = \frac{1}{(t_2 - t_1)^2} \sum_{j=t_1}^{t_2-1} \sum_{k=t_1}^{t_2-1} a_{jk}$$

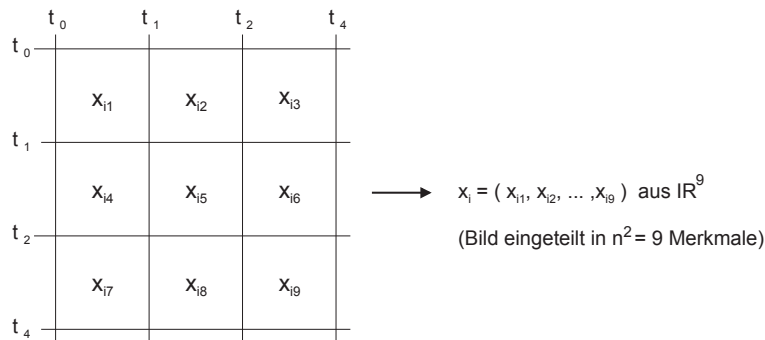


Abbildung 6: Merkmalsextraktion aus  $32 \times 32$ -Pixelbild

### 3.5 Formulierung des Parameterbestimmungsproblems

Aus den bisherigen Überlegungen wird versucht das Problem der Grauwertbestimmung formal zu beschreiben. Es zeigt sich (s. Ergebnisse Abschnitt 5), dass erst bei  $n \geq 3$  (d.h. Merkmalsvektoren ab der Dimension 9), eine gewisse Generalisierbarkeit der Methoden erreicht werden kann. Für  $n = 1$  und  $n = 2$  sind die Informationen in den Merkmalen nicht genügend differenzierbar, da die Informationen über Hintergrund und Objekt zu sehr vermischt sind. Da der funktionale Zusammenhang schwierig zu erkennen ist, ist ein neuronales Netz ein geeigneter Ansatz zur Lösung dieses Problems. Im nächsten Abschnitt werden die Methoden vorgestellt die sich für dieses Problem eignen. Gesucht ist eine Funktion die folgende Abbildung realisiert:

$$f : \mathbb{R}^d \mapsto \mathbb{R} \quad (10)$$

$$f(x_i) = \hat{y}_i \quad 0 \leq \hat{y}_i \leq G - 1, \hat{y}_i \in \mathbb{R}, x_i \in \mathbb{R}^d \quad (11)$$

Ein Merkmalsvektor  $x_i \in \mathbb{R}^d$  (welcher das Muster  $i$  repräsentiert) wird auf einen Wert im Bereich von 0 bis  $G - 1$  abgebildet. Dabei gibt  $G$  den Grauwertbereich an (hier  $G = 256$ ). Weiter werden hier nur ganzzahlige Grauwerte betrachtet. Deshalb wird das Ergebnis  $\hat{y}_i$  der Funktion nach den üblichen Rundungsregeln auf- oder abgerundet.

## 4 Methoden

In diesem Abschnitt werden die Methoden vorgestellt, die sich für das Problem der Parameterbestimmung eignen. Zunächst wird die bisher verwendete sogenannte 60%-Methode erläutert. Als erster Ansatz wird die Parameterbestimmung mittels Abstandsklassifikatoren untersucht. Eine recht einfache Methode ist das  $k$ -Nearest-Neighbour-Verfahren. Hierbei werden die Nachbarn mit den kleinsten Abständen gesucht. Eine allgemeinere Variante von Nearest-Neighbour ist die Methode der Kernel-Densities. Die Nachbarn werden dabei nach ihrem Abstand gewichtet. Der Nachbar mit dem höchsten Gewicht bekommt den größten Faktor für die Bestimmung des Grauwertparameters. Als letzte Methode werden neuronale Netze untersucht. Bei diesen Verfahren ist eine Datenbank mit genügend Beispielbildern erforderlich. Für die Untersuchung wurde jedoch nur eine kleine Datenbank mit 82 Beispielbildern zur Verfügung gestellt. Eine weitere Datenbank mit 608 Bildern wurde ergänzt.

Im Allgemeinen wird die Güte eines Klassifikators über die Fehlerrate ermittelt. Bei Klassifikationsproblemen hat man i.d.R. nur „Richtig“ oder „Falsch“ Entscheidungen bezüglich eines klassifizierten Bildobjektes. Bei Regressionsproblemen wird es auch möglich einen absoluten Fehler anzugeben. Als Güte wird hier die Standardabweichung benutzt

Die hier benutzten Abstandsklassifikatoren und das neuronale Netz durchlaufen eine Trainings- und Testphase, wobei das Training mittels „leaving-one-out“ durchgeführt wird. Das bedeutet, dass das Training beispielsweise ohne Muster  $i$  stattfindet. Ist das Training beendet, so wird für  $x_i$  der geschätzte Grauwert ermittelt. Ist die Ermittlung des Grauwertparameters abhängig von Trainingsdaten so kann die Berechnung allgemein wie folgt formuliert werden. Sei die Trainingsmenge  $T$  ohne  $x_i$  gegeben durch

$$T \setminus \{x_i\} = \{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\},$$

so wird der geschätzte Grauwert für Muster  $i$  bestimmt durch

$$\hat{y}_i = F(x_i, T \setminus \{x_i\})$$

Im Unterschied zu (5) soll mit  $F$  verdeutlicht werden, dass eine Trainingsphase durchlaufen werden muss. Ist für jedes Trainingsdatum  $x_i$  der geschätzte Wert  $\hat{y}_i$  berechnet, dann ist die mittlere quadratische Abweichung (MSE, mean square error) definiert durch

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

und die Standardabweichung mit  $\text{RMSE} = \sqrt{\text{MSE}}$  (root mean square error).

Ein weiterer Aspekt im Kontext der Grauwertbestimmung ist die Normalisierung. Ziel ist es, dass die Grauwertbestimmung unabhängig von der Helligkeit des Bildes funktioniert, daher wird zunächst normalisiert (12) und der Ergebniswert nach der Regression entsprechend zurückgerechnet (13). Eine wichtige Nebenbedingung ist z.B. dass bei genau gleichem Objekt der Ergebniswert direkt proportional zur Gesamthelligkeit ist. Für die Normierung der Merkmalsvektoren wurde die euklidische Norm verwendet, die wie folgt definiert ist:

$$\|x_i\| = \sqrt{\sum_{j=1}^d x_{ij}^2}$$

Der normalisierte Merkmalsvektor ergibt sich dann durch

$$x_i^{\text{norm}} = x_i \cdot \frac{1}{\|x_i\|} \quad (12)$$

Sei  $T^{\text{norm}}$  die Menge der normalisierten Merkmalsvektoren mit

$$T^{\text{norm}} = \{x_1^{\text{norm}}, \dots, x_n^{\text{norm}}\}$$

Der Ergebniswert ergibt sich dann durch Umkehrung der Normalisierung durch

$$\hat{y}_i = \|x_i\| \cdot F(x_i^{\text{norm}}, T^{\text{norm}} \setminus \{x_i^{\text{norm}}\}) \quad (13)$$

#### 4.1 Die 60%-Methode

Der Hintergrund für die Benutzung der 60%-Methode ist die Tatsache, dass die wesentliche Information im Röntgenbild sehr häufig im Zentrum liegt. Wie bei der Spotmessung in der Fotografie (vgl. Abschnitt 2) wird der Grauwert mittels der Werte aus dem 60%-Bereich des Bildzentrums berechnet, in der Hoffnung, dass dieser geschätzte Grauwert dem optimalen Grauwert sehr nahe kommt. Der optimale Grauwert, der bei den 82 Trainingsdaten per Hand ermittelt wurde, ist ein Wert, der das Objekt in bestmöglichem Kontrast darstellt.

Die Schritte für Bestimmung des Grauwertes durch die 60%-Methode sind in Abb. 7 dargestellt. Die Merkmalsanalyse besteht darin, dass aus den 1024 Grauwerten des Musters  $i$  alle die Grauwerte bestimmt werden, die sich in dem 60%-Bereich befinden. Mittels geeigneter Verfahren kann der Bildausschnitt  $S$  bestimmt werden. Der geschätzte Grauwert für Muster  $i$  ergibt sich dann aus dem Durchschnitt der Grauwerte in  $S$ .

$$f_{60\%}(x_i) = \frac{1}{|S|} \sum_{x_{ik} \in S} x_{ik}$$

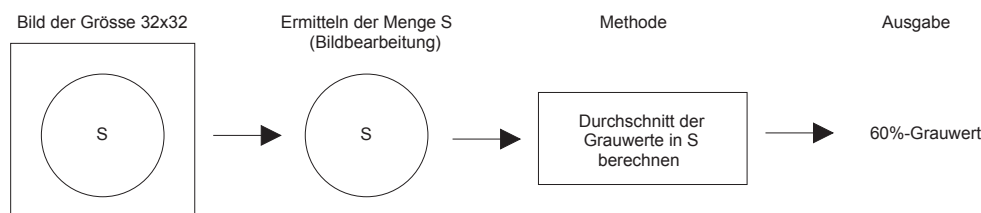


Abbildung 7: Schritte bei der 60%-Methode

Bei der Implementierung dieser Methode ist es aus Effizienzgründen sinnvoll, Merkmalsanalyse und Durchschnittberechnung in einem Schritt durchzuführen. In dieser Arbeit dient diese Methode jedoch nur als Referenz für die anderen Methoden. Bei der kleinen Datenbank sind die Ergebnisse der 60%-Methode neben den per Hand ermittelten Werten in den zur Verfügung gestellten Daten aufgeführt. Bei der großen Datenbank sind diese Werte jedoch nicht bestimmt worden. Um auch hier einen Vergleich zu haben, wurden die 60%-Werte näherungsweise bestimmt, wie in Abb. 8 skizziert ist. Hierbei werden nur quadratische Felder betrachtet. Dabei umschließt die Menge  $P_1$  mit einer Größe von  $26 \times 26$  Grauwerten die Menge  $S$  und ist anteilmäßig 66%.  $P_4$  ist die kleinste Menge mit Größe  $20 \times 20$  und Anteil 39% und ist in  $S$  enthalten.  $P_2$  und  $P_3$  liegen dazwischen. Der approximierte Grauwert wird dann wie folgt berechnet.

$$f_{\widehat{60\%}}(x_i) = \frac{1}{4} \sum_{j=1}^4 \left( \frac{1}{|P_j|} \sum_{x_{ik} \in P_j} x_{ik} \right)$$

Die Ergebnisse dieser Näherungsmethode wurden, anhand einer Auswahl von Bildern der kleinen Datenbank, mit den richtigen Werten verglichen. Dabei wurde auch jedes  $p_i$  einzeln mit dem richtigen Wert verglichen. Es zeigte sich jedoch, dass der Durchschnitt die beste Näherung erzielte. Die Abweichung von den tatsächlichen Werten war so gering, dass auf eine weitere Verfeinerung verzichtet wurde.

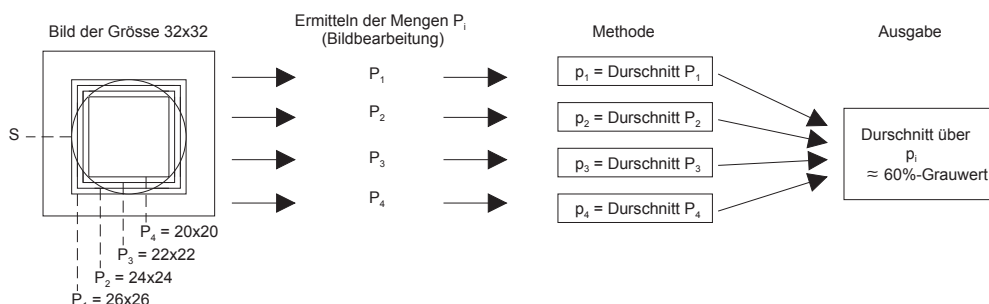


Abbildung 8: Vergleichsmethode zum 60%-Grauwert

## 4.2 Nearest-Neighbour

Die *nearest neighbour decision rule* oder kurz NN-Regel ist ein einfaches Verfahren aus der Mustererkennung. Die NN-Regel sucht zu einem gegebenem Muster  $i$  den nächsten Nachbarn aus der Menge der Trainingsdaten. Sei Muster  $j$  der nächste Nachbar von  $i$ , dann wird  $i$  die Eigenschaft von  $j$  zugewiesen. Im Kontext der Grauwertbestimmung wird Muster  $i$  den Grauwert von  $j$  zugewiesen. Der  $k$ -nearest-neighbour ( $k$ -NN) ist die allgemeine Form dieses Klassifikators. Hierbei werden die  $k$  nächsten Nachbarn zur Entscheidung herangezogen und beispielsweise der Durchschnitt der Grauwerte dieser ermittelten  $k$  Nachbarn berechnet. Hier werden nur die Varianten  $k = 1$  und  $k = 3$  betrachtet. Die Nearest-Neighbour Regel ist durch folgende Abbildung gegeben:

$$r : \mathbb{R}^d \rightarrow \{1, \dots, K\}$$

$$r(x) = k_{\hat{i}}, \quad \hat{i} = \operatorname{argmin}_i \{d(x, x_i)\} \quad x, x_i \in \mathbb{R}^d$$

Als Abstandsmaß für  $d(x, y)$  wird hier der euklidische Abstand (6) benutzt. Zu einem gegebenen Testvektor  $x \in \mathbb{R}^d$  wird mittels der Trainingsdaten  $x_1, \dots, x_n$  der Nachbar mit kleinstem euklidischem Abstand ermittelt. Die NN-Regel wird i.d.R. für Klassifikationsaufgaben eingesetzt, wie z.B. bei der Ziffernerkennung. Dabei gibt es  $\{0, \dots, 9\}$  Zielklassen. Ist nach der NN-Regel  $x_i$  der nächste Nachbar von  $x$  dann wird  $x$  in die Klasse zugewiesen mit der  $x_i$  gelabelt ist.

Im Kontext der Grauwertbestimmung wird die NN-Regel hier wie folgt verwendet. Sei  $x_i$  der nächste Nachbar von  $x$ , dann ist der geschätzte Grauwert von  $x$  gleich dem optimalen Grauwert von  $x_i$  mit

$$f_{\text{NN}} : \mathbb{R}^d \mapsto \mathbb{R}$$

$$f_{\text{NN}}(x) = y_{\hat{i}}, \quad \hat{i} = \operatorname{argmin}_i \{d(x, x_i)\} \quad x, x_i \in \mathbb{R}^d$$

Beim 3-NN wird der Durchschnitt der Grauwerte der drei nächsten Nachbarn  $x_i, x_j, x_k$  bestimmt. In Abb. 8 wird die NN-regel veranschaulicht. In diesem Beispiel wäre der nächste Nachbar von Testdatum  $x$  das Trainingsdatum  $x_3$ , womit der geschätzte Wert  $\hat{y} = y_3$  wäre. Der 3-NN würde die Trainingsdaten  $x_3, x_4$  und  $x_5$  bestimmen wodurch sich als geschätzte Wert  $\hat{y} = (y_3 + y_4 + y_5)/3$  ergeben würde.

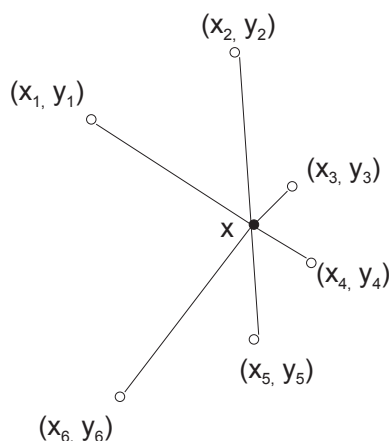


Abbildung 9: Veranschaulichung der NN-Regel

### 4.3 Kernel-Densities

Die Kernel-Densities (KD) werden z.B. bei statistischen Modellierungen wie z.B. bei der statistischen Klassifikation von Einzelobjekten eingesetzt [8] [9]. Die anschauliche Idee dieses Ansatzes ist es, dass die Klassifikation nicht nur über den nächsten Nachbarn entschieden wird (NN-Regel), sondern dass möglichst viele Trainingsdaten zum Ergebnis beitragen. Ein optionaler Parameter der Kernel-Densities ist die Wahl einer geeigneten Kernfunktion. Hier wird eine Gaußverteilung als Kernfunktion verwendet, welche radial-symmetrisch ist.

Aufgrund der Gaußverteilung wird es möglich, die Trainingsdaten nach ihrem euklidischen Abstand so zu gewichten, dass der Nachbar mit dem kleinsten Abstand das höchste Gewicht bekommt, der nächste Nachbar das zweithöchste Gewicht usw. Der Nachbar mit dem kleinsten Abstand trägt mit dem größten Anteil und die nächsten Nachbarn tragen mit immer kleiner werdenden Anteilen zum Ergebnis bei. Abbildung 10 skizziert ein Beispiel. Trainingsdatum  $x_1$  hat den kleinsten Abstand zum Testdatum  $x$  und deshalb wird der zu bestimmende Grauwert mit größtem Anteil von  $y_1$  bestimmt. Da  $x_6$  den größten Abstand aufweist, wird  $y_6$  nur minimal zum Ergebnis beitragen. Die KD-Regel entspricht der folgenden Abbildung

$$f_{\text{KD}} : \mathbb{R}^d \mapsto \mathbb{R}$$

$$f_{\text{KD}}(x) = \left( \sum_{i=1}^n \exp\left(-\frac{d(x, x_i)}{2 \sigma^2 \alpha}\right) \right)^{-1} \sum_{i=1}^n y_i \cdot \beta_i \cdot \exp\left(-\frac{d(x, x_i)}{2 \sigma^2 \alpha}\right) \quad (14)$$

Als Abstandsmaß für  $d(x, y)$  wird hier der (quadrierte) euklidische Abstand (6) benutzt. Der Faktor  $\beta_i$  ist der Normalisierungsfaktor des Merkmalsvektors  $x_i$ . Wird nicht normalisiert dann gilt  $\beta_i = 1$ , sonst ist  $\beta_i = 1/\|x_i\|$ .  $n$  ist die Anzahl der Trainingsdaten. Aufgrund der Art der Trainingsdaten kommt es vor, dass die Varianzen falsch geschätzt werden. Mit dem Faktor  $\alpha$  kann die Varianz gesteuert und solche Abweichungen kompensiert werden. Die Varianz  $\sigma^2$  ist gegeben durch

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n d(x_i, \bar{x})$$

und der Mittelwertsvektor  $\bar{x}$  mit

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Im Normalfall tragen alle Trainingsdaten  $x_1, \dots, x_n$  zur Schätzung des Grauwertes für das Testdatum  $x$  bei. Es ist jedoch zu bemerken, dass die Gewichte von Trainingsdaten mit großem euklidischen Abstand sehr klein werden und damit zum Ergebnis kaum noch beitragen. Das bedeutet, dass üblicherweise weniger als  $n$  Trainingsdaten signifikant zum Ergebnis beitragen.

Der Nenner im Quotient des Exponenten in (14) kann durch die Bestimmung von  $\alpha$  in die folgende Grenzen laufen

$$\alpha \sigma^2 \rightarrow 0 \implies f_{\text{KD}}(x) \rightarrow y_{\hat{i}} \quad \text{mit } \hat{i} = \underset{i}{\operatorname{argmin}} \{d(x, x_i)\} \quad (15)$$

$$\alpha \sigma^2 \rightarrow \infty \implies f_{\text{KD}}(x) \rightarrow \frac{1}{n} \sum_{i=1}^n y_i \quad (16)$$

Wird  $\alpha$  sehr klein gewählt (d.h.  $\alpha \rightarrow 0$ ), so wird die Gaußverteilung so weit gestaucht, dass der geschätzte Grauwert nur noch durch den nächsten Nachbarn bestimmt wird. Alle anderen



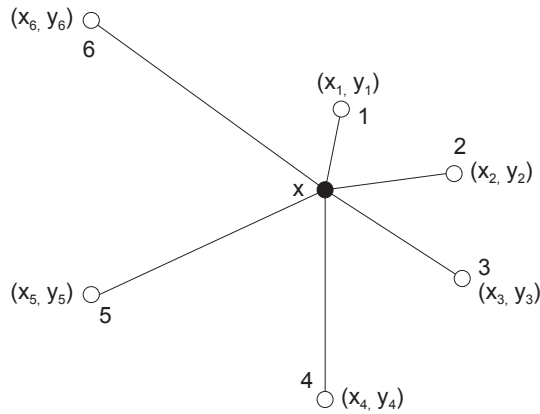


Abbildung 10: Veranschaulichung der KD-Regel

Trainingsdaten haben relativ kleine Gewichte, so dass sie nicht signifikant zum Ergebnis beitragen. In diesem Fall ist das Verfahren identisch zum Nearest-Neighbour Verfahren. Wird  $\alpha$  groß (d.h.  $\alpha \rightarrow \infty$ ), so ergibt sich der geschätzte Grauwert aus dem Durchschnitt der Grauwerte aller Trainingsdaten (16). Im letzteren Fall bedeutet dies, dass alle Trainingsdaten signifikant zum Ergebnis beitragen.

In der Trainingsphase wird versucht, den Parameter  $\alpha$  möglichst gut einzustellen, wodurch gleichzeitig ein Optimierungsproblem gegeben ist. Je nach Problem, kann ein großes oder auch ein kleines  $\alpha$  geeignet sein. Bei der Grauwertbestimmung erscheint es sinnvoll, dass nur eine kleine Zahl der nächsten Nachbarn das Ergebnis bestimmen sollten. Ein großes  $\alpha$  hat zur Folge, dass (14) zum Durchschnitt der Grauwerte (16) tendiert. Es zeigte sich auch in den Testläufen, dass diese Einstellung zu schlechteren Ergebnissen führt. Mit gewissen Vorüberlegungen kann der Bereich von  $\alpha$  eingeschränkt werden und schliesslich schneller gefunden werden. Hier wurde der Wert von  $\alpha$  experimentell bestimmt. Hierzu wurden Testläufe mit verschiedenen  $\alpha$  durchgeführt.

Bei der Implementierung der KD-Regel muss darauf geachtet werden, dass zur Laufzeit keine numerischen Probleme auftreten. Diese können dadurch entstehen, dass die Gewichte der Trainingsdaten sehr klein werden (die Gewichte von Trainingsdaten die einen zu großen euklidischen Abstand zum Testdatum haben). Es treten dann sogenannte *floating point* (hier: *underflow*) Fehler auf, da die Zahlen nicht mehr mit den üblichen Datentypen im Rechner dargestellt werden können. Um dieses Problem zu vermeiden, berechnet man zunächst alle euklidischen Distanzen und bestimmt das Minimum  $d_{\min}$  unter diesen. Die KD-Regel bekommt nun die folgende Form.

$$f_{\text{KD}}(x) = \left( \sum_{i=1}^n \exp\left( - \frac{d(x, x_i) - d_{\min}}{2 \sigma^2 \alpha} \right) \right)^{-1} \sum_{i=1}^n \text{opt}(x_i) \cdot \beta_i \cdot \exp\left( - \frac{d(x, x_i) - d_{\min}}{2 \sigma^2 \alpha} \right) \quad (17)$$

Durch diese Methode bleiben die exponentiellen Terme im Rahmen des darstellbaren, so dass keine numerischen Probleme auftreten.

#### 4.4 Neuronale Netze

Künstliche neuronale Netze sind in Anlehnung an das biologische Vorbild - das menschliche Gehirn - entstanden. Zur Unterscheidung von biologischen neuronalen Netzen spricht man i.A. von künstlichen neuronalen Netzen. Das biologische Neuron besteht aus drei Komponenten: dem Zellkörper, den Dendriten und einem Axon. Die Dendriten sind Verbindungen über die Synapsen und summieren die Eingabe des Netzes in die Zelle auf. Die eigentliche Funktion der Zelle

wird durch das Axon gebildet, wobei jede Zelle genau ein Axon hat. Diese leitet die Ausgabe der Zelle weiter an folgende Synapsen. Mit einem numerischen Wert kann die Stärke einer Synapse (Verbindungsgewicht) beschrieben werden [10]. Ein künstliches neuronales Netz besteht aus stark idealisierten Neuronen. Eine systematische Einführung befindet sich etwa in [13].

#### 4.4.1 Das Multilayer-Perceptron

In dieser Arbeit wird nur ein spezielles neuronales Netz betrachtet, das sogenannte Multilayer-Perceptron (MLP). Ein MLP ist ein gewichtetes Netz und besteht aus mehreren Schichten. Jede Schicht enthält eine Menge von Knoten - das sind die Neuronen - und jedes Neuron arbeitet parallel und unabhängig von allen anderen. Bezogen auf die Topologie sind MLPs Netze ohne Rückkopplung (feedforward-Netze), d.h es existiert kein Pfad von einem Neuron direkt oder über zwischengeschaltete Neuronen wieder zurück zu diesem Neuron. Bei feedforward-Netzen werden die Daten immer nur in eine Richtung weitergegeben. Formal entspricht diese Topologie einem azyklischen Graphen. Eine weitere Eigenschaft von MLPs ist, dass sie ebenenweise verbundene Netze sind, d.h dass es nur Verbindungen von einer Schicht zur nächsten gibt. Man spricht von einem vollständigen feedforward Netz, falls jedes Neuron der Schicht  $i$  mit jedem Neuron der Schicht  $i + 1$  verbunden ist.

In Abb. 11 wird ein MLP für Klassifikationsaufgaben gezeigt. Die Eingangsschicht  $S_{in}$  enthält genauso viele Neuronen wie die Dimension des Merkmalsvektors (hier im Beispiel  $d = 3$ ), d.h. für jede Komponente ein Eingabeneuron. Die arbeitenden Schichten sind die verdeckten Schichten  $V_1$  und  $V_2$ . Seien  $1, \dots, K$  Zielklassen vorhanden, dann gibt es pro Klasse einen Neuron in der Ausgangsschicht  $S_{out}$ . Ein Merkmalsvektor  $x = (x_{i1}, x_{i2}, x_{i3})$  wird in das Netz eingegeben und an den Ausgängen können die Ausgangswerte entnommen werden. Die Ausgangswerte können auf verschiedene Art und Weise interpretiert werden. Bei Klassifikationsaufgaben könnte man beispielsweise die Ausgänge  $k_1, k_2$  und  $k_3$  als Scores betrachten. Der Ausgang mit dem höchsten Score gibt an, zu welcher Klasse das Muster klassifiziert wird.

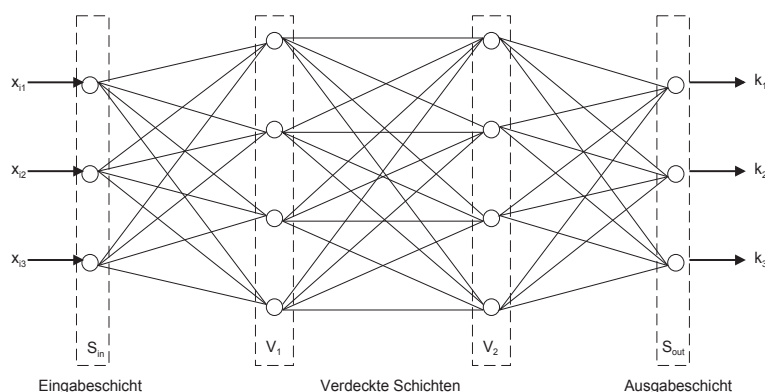


Abbildung 11: Vollständig verbundenes MLP mit zwei verborgenen Schichten

Die arbeitenden Elemente eines neuronalen Netzes sind die Neuronen selbst. Der Aufbau eines Neurons wird in Abb. 12 gezeigt. Seien  $u_1, \dots, u_L$  die Eingänge an einem Neuron. Die Eingänge werden gewichtet aufsummiert und bilden damit die Eingabe für die sogenannte Aktivierungsfunktion  $h$ . Die Ausgabe des Neurons ist das Ergebnis von  $h$  mit

$$u_i = h\left(\sum_{l=0}^L w_{il} \cdot u_l\right), \quad u_0 = 1$$

und gleichzeitig Eingabe für Neuronen der nächsten Schicht. Die zusätzliche Eingabe eines konstanten Faktors  $u_0 = 1$  gewährleistet die vollständige Freiheit bei der Definition von Trennflächen zur Klassifikation. Die Aktivierungsfunktion kann prinzipiell für jedes Neuron separat gewählt werden, i.d.R. wird aber nur eine Funktion verwendet. Als Aktivierungsfunktionen können Schwellwert- oder auch Sigmoid-Funktionen eingesetzt werden, wodurch diskrete oder kontinuierliche Wertebereiche möglich sind, wobei der Wertebereich  $[0, 1]$  (oder  $[-1, 1]$ ) die Regel ist. Bei Schwellwertfunktionen wird das Ergebnis wie folgt ermittelt.

$$h(a) = \begin{cases} 0 & , \text{ falls } a < S \\ 1 & , \text{ falls } a \geq S \end{cases}$$

Für kontinuierliche Wertebereiche in den oben genannten Intervallen wird beispielsweise die monoton wachsende und differenzierbare sigmoide Funktion  $f(x) = \frac{1}{1+e^{-x}}$  gewählt. Die genannten Eigenschaften der sigmoiden Funktion sind notwendig, wenn das MLP mittels Lernverfahren wie Backpropagation-Learning trainiert werden soll.

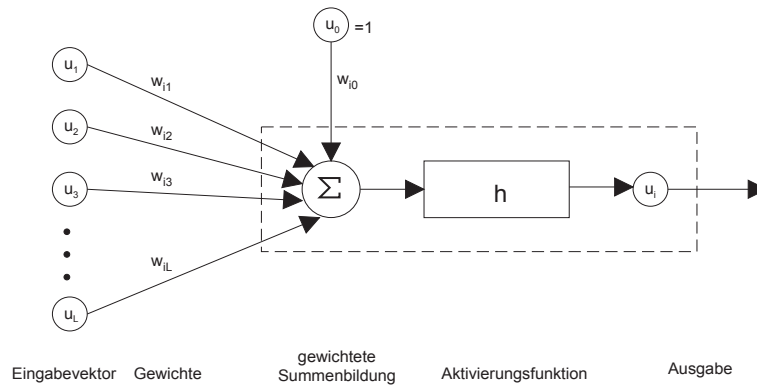


Abbildung 12: Schematischer Aufbau eines Neurons

Die Funktionsweise eines neuronalen Netzes kann in zwei Phasen eingeteilt werden [9]. In der Ruhephase sind alle Aktivierungen, d.h. die Neuronenausgänge, konstant. Wird ein Merkmalsvektor eingegeben, dann beginnt die Arbeitsphase. Jede Schicht wird nacheinander durchlaufen und jedes Neuron berechnet seine Aktivierung neu, bis das Netz wieder in die Ruhephase gelangt. Das Lernen eines neuronalen Netzes geschieht dadurch, dass es gemäß einer vorgegebenen Vorschrift - der Lernregel - modifiziert wird. Die Lernregel (nach [10]) ist ein Algorithmus, nach dem das Netz lernt, für eine vorgegebene Eingabe eine gewünschte Ausgabe zu produzieren. Durch wiederholte Eingabe von Trainingsmustern werden die Gewichte der Verbindungen modifiziert (18). Dabei wird versucht, den Fehler zwischen erwarteter und tatsächlicher Ausgabe zu minimieren, d.h. eine optimale Konfiguration der Gewichte zu finden. Wie schon in Abschnitt 3 beschrieben worden ist, wird hier als Trainingsart das überwachte Lernen (supervised learning) verwendet. Die optimalen Werte (hier die Grauwerte) der Trainingsdaten sind bekannt. Die Gewichtsveränderung des Gewichtes  $w_{ij}$  wird nach der delta-Regel wie folgt durchgeführt.

$$w_{ij}^{\text{neu}} = w_{ij}^{\text{alt}} + \Delta w_{ij} \quad , w_{ij} \in \mathbb{R} \quad (18)$$

Die Lernregel lautet dann für den Gewichtsvektor (20)

$$w^{\text{neu}} = w^{\text{alt}} - \eta \Delta w C(w) \quad w \in \mathbb{R}^q, \eta \in \mathbb{R} \quad (19)$$

Dabei bezeichnet  $-\Delta w C(w)$  den negativen Gradienten und  $\eta$  die Lernrate. Bei jeder Korrektur wird der Gewichtsvektor um den Bruchteil  $\eta$  des Gradienten verändert. Jeder Punkt auf der

Fehleroberfläche besitzt eine Neigung in Richtung des nächstliegenden Tals. Der Gewichtsvektor wird in jedem Schritt in Richtung Tal geschoben, um ein Minimum zu erreichen. Die Berechnung des Gradienten wird über partielle Ableitungen mittels der Kettenregel durchgeführt. Eine Herleitung befindet sich etwa in [10].

Das Lernverhalten eines neuronalen Netzes wird über die Fehler gesteuert. Allgemein gilt, dass das Backpropagation-Netz für einen Eingabevektor  $x$  und einer vorgegebenen Konfiguration der Gewichte, immer die selbe Ausgabe liefert. Sei  $q$  die Gesamtzahl der Gewichte, dann ist der Gewichtsvektor des Netzes gegeben durch

$$w = \{w_1, \dots, w_q\} \quad \text{mit } w \in \mathbb{R}^q \quad (20)$$

Die Ausgabe eines MLP kann deshalb als Funktion über den Eingabevektor und dem Gewichtsvektor beschrieben werden.

$$\begin{aligned} f_{\text{MLP}} &: \mathbb{R}^d \times \mathbb{R}^q \mapsto \mathbb{R} \\ \hat{y} &= f_{\text{MLP}}(x, w) \quad \text{mit } w \in \mathbb{R}^q \end{aligned}$$

Wie schon beschrieben worden ist, wird die Abweichung des Grauwertes für ein Trainingsdatum  $(x, y)$  durch  $\hat{y} - y$  angegeben. Die mittlere quadratische Abweichung (MSE) wird für das neuronale Netz als Funktion über die Gewichte beschrieben durch:

$$E(w) = \frac{1}{n} \sum_{i=1}^n (f_{\text{MLP}}(x_i, w) - y_i)^2 = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

Durch variieren von  $w$  lässt sich  $E(w)$  für den ganzen Gewichtsraum bestimmen. Sei für alle Gewichtskombinationen der MSE bekannt, kann der Fehler gegen die Gewichte in ein Koordinatensystem eingetragen werden. In Abbildung 13 wird eine vereinfachte Darstellung einer Fehlerverlaufskurve gezeigt. Dabei sei  $W$  als Vektorraum aller  $w$  definiert. Das eigentliche Ziel bei einem neuronalen Netz besteht darin, die Konfiguration der Gewichte zu finden, bei der das Netz gut generalisiert. Es wird erhofft, dass dieses Ziel dadurch erreicht wird, indem die Konfiguration der Gewichte gefunden wird, welche  $E$  minimiert. Ein zu langes Training wird dazu führen, dass das Netz die Trainingsdaten zu gut lernt. Eine mögliche Folge ist es, dass unbekannte Testdaten sehr schlecht klassifiziert werden. Wird das Training zu schnell abgebrochen, dann hat das Netz noch nicht genug gelernt, sodass es schlecht generalisiert. Die Lösung dieses Problems liegt darin, dass das Netz in verschiedenen Trainingsphasen getestet wird, um die beste Generalisierung zu finden. Es ist an dieser Stelle zu bemerken, dass die Berechnung der

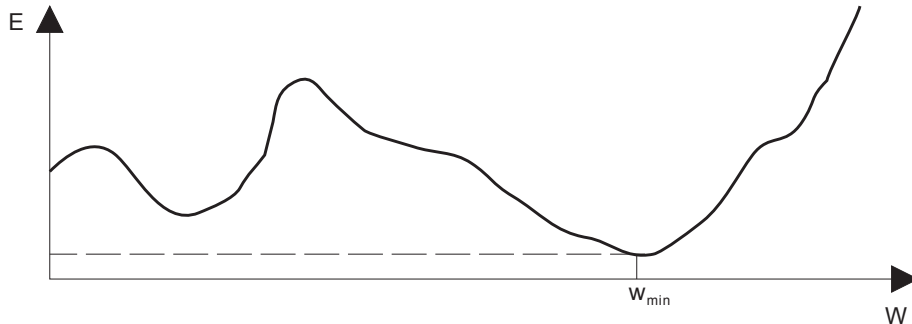


Abbildung 13: Mögliche vereinfachte Struktur einer Fehleroberfläche

gesamten Fehleroberfläche vom Rechenaufwand inakzeptabel groß wäre. Der Gradientenabstieg ist ein Verfahren, das das Minimum von  $E$  findet, ohne die gesamte Fehleroberfläche zu kennen.

Dieses wird im folgenden erklärt.

Das Backpropagation-Verfahren beruht auf einem Gradientenabstiegsverfahren auf der Fehlerfunktion. Dabei sucht der Algorithmus ein Minimum der Fehlerfunktion durch Abstieg in der Gradientenrichtung entlang der Fehlerfunktion. Dazu startet das Verfahren mit einem zufällig gewählten Gewichtsvektor  $w$ . In diesem Punkt wird der Gradient bestimmt und um eine vorgegebene Länge, die Lernrate, herabgestiegen. Die so erhaltene Gewichtsveränderung liefert den Gewichtsvektor, in dem wiederum der Gradient bestimmt wird. Das Verfahren wiederholt sich, bis ein lokale Minimum erreicht ist [10]. Die Konfiguration der Gewichte, die den Berechnungsfehler minimiert, wird als Lösung des Lernproblems betrachtet. Eine notwendige Voraussetzung der Fehlerfunktion ist, dass der Gradient für alle Punkte des Gewichtsraum existiert, d.h. die partiellen Ableitungen der Fehlerfunktion nach den einzelnen Gewichten müssen überall definiert sein. Aus diesem Grund werden in der Regel sigmoide Funktionen als Aktivierungsfunktion verwendet, wodurch die Fehlerfunktion stetig und überall differenzierbar ist [13]. Ein weiterer Vorteil ist es, dass die Ableitung einfach zu berechnen ist durch

$$f'(x) = \left( \frac{1}{1 + e^{-x}} \right)' = \frac{e^{-x}}{(1 + e^{-x})^2} = f(x)(1 - f(x)).$$

Weitere Vorteile einer sigmoiden Funktion sind, dass die Struktur der Fehleroberfläche durch sigmoide Funktionen wesentlich beeinflusst wird. Die Fehleroberfläche wird geglättet und erleichtert damit den Gradientenabstieg.

An dieser Stelle sollen einige Probleme erläutert werden, die das Verfahren des Gradientenabstiegs mit sich bringt. Da der Gradientenabstieg mit einer zufällig gewählten Konfiguration der Gewichte startet, kann nicht festgelegt werden, ob ein lokales oder globales Minimum gefunden wird. Lag  $w$  in der Nähe eines globalen Minimums, endet der Gradientenabstieg unmittelbar in dem globalen Minimum. Lag  $w$  hingegen in der Nähe eines lokalen Minimums, so wird es im lokalen Minimum enden. Mit wachsender Dimension des Netzes wird die Fehleroberfläche immer stärker zerklüftet, wodurch die Anzahl der lokalen Minima steigt [10]. Damit wird es dem Verfahren weiter erschwert, ein globales Minimum zu erreichen. Ein anderes Problem tritt bei vollständig verbundenen Feedforward-Netzen auf. Bei dieser Art von Netzen dürfen die Gewichte nicht alle gleich initialisiert werden. Dies hätte zur Folge, dass alle Gewichte, die von einem Eingabeneuron ausgehen bzw. ein Ausgabeneuron erreichen, auch nach der Gewichtsmodifikation stets den gleichen Wert aufweisen. Das Problem wird dadurch gelöst, dass die Gewichte zufällig initialisiert werden. Die Größe der Gewichtsveränderung wird bei Gradientenabstiegsverfahren maßgeblich durch den Betrag des Gradienten bestimmt. In flachen Plateaus wird der Gradient sehr klein und das Verfahren stagniert nahezu. Ein weiteres Problem ist die Oszillation. Bei einer unglücklichen Konstellation von Lernrate und Fehleroberfläche kann das Verfahren oszillieren.

Die oben genannten Probleme erfordern ein tieferes Studium von neuronalen Netzen und würden den Rahmen dieser Arbeit sprengen. Das Ziel dieser Arbeit ist es, die Grauwertbestimmung mittels neuronalen Netzes zu untersuchen. Als Ansatz wurde hierfür ein relativ einfaches Netz gewählt, welches im folgenden beschrieben wird.

#### 4.4.2 Neuronales Netz für die Parameterbestimmung

In diesem Abschnitt wird das neuronale Netz beschrieben, welches für die Parameterbestimmung gewählt wurde. Wie in Abschnitt 3 beschrieben worden ist, ist eine Netzstruktur zu wählen, die das nichtlineare Regressionsproblem der Grauwertbestimmung löst. Die Bestimmung der Anzahl der Schichten und der Anzahl der Neuronen kann bei neuronalen Netzen i.A. nicht analytisch hergeleitet werden. Das bedeutet, dass die richtige Lösung experimentell gefunden werden muss.

Es gibt jedoch grobe Regeln für die Bestimmung der Anzahl der Schichten eines neuronalen Netzes [11], welche sich aus der Komplexität des vorliegenden Problems ergibt. Für linear trennbare Klassen reicht sogar ein lineares Single-Layer-Perzeptron aus, d.h. das neuronale Netz besteht nur aus einer Schicht. Sind die Klassen nicht mehr linear trennbar, so werden neuronale Netze mit einer verdeckten Schicht verwendet. Netzwerke mit größerer Anzahl von Schichten werden verwendet, um beliebige Trennflächen zu modellieren, oder um Vorwissen einzubringen.

Nach den theoretischen Überlegungen in Abschnitt 3, handelt es sich bei der Grauwertbestimmung um ein nichtlineares Regressionsproblem. Da die Komplexität des Problems nicht einfach zu bestimmen ist, wurde als erster Ansatz, ein neuronales Netz mit einer verdeckten Schicht gewählt. In Abbildung 14 wird das verwendete Neuronale Netz gezeigt. Die Anzahl der Neuronen in der verdeckten Schicht wurde variabel gelassen. Die Experimente wurden mit 10 bis 50 Neuronen in der verdeckten Schicht durchgeführt. Die Ausgangsschicht besteht aus genau einem Neuron, da nur ein Wert ausgegeben wird. Jede Komponente  $x_{ij}$  eines Merkmalsvektors  $x_i$  wird in jedes Neuron der Eingangsschicht eingegeben. Die Ausgaben der Neuronen  $p_1^{(2)}, \dots, p_L^{(2)}$  der verdeckten Schicht bilden die Eingabe des einzigen Neurons der Ausgangsschicht. Dieses produziert schliesslich die Ausgabe des Neuronalen Netzes.

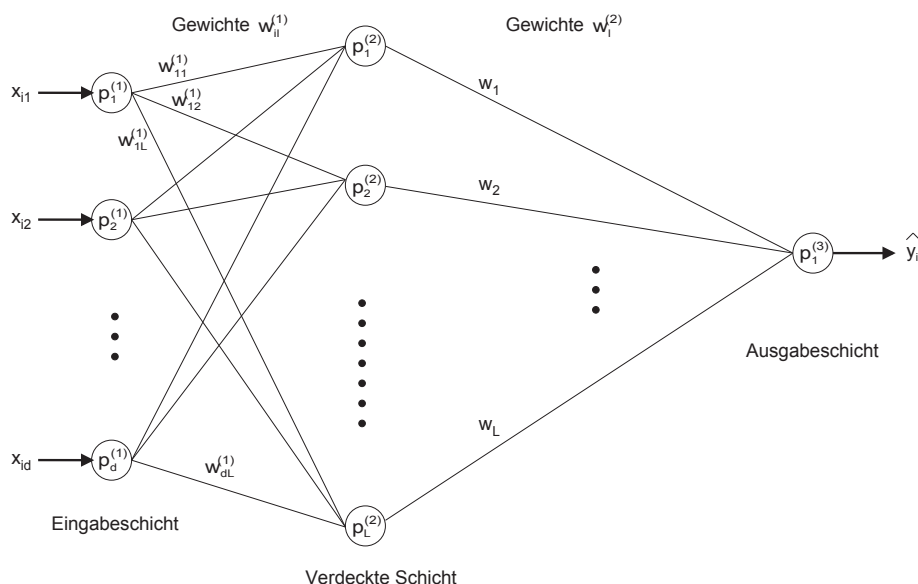


Abbildung 14: Hier verwendetes Neuronales Netz mit Eingangs- und Ausgangsschicht, einer verdeckten Schicht. Die Zahl der Neuronen in der verdeckten Schicht wurde in den Experimenten variiert.

Aufgrund der Art der Daten ist eine Vorverarbeitung der Merkmale notwendig gewesen, bevor die Merkmalsvektoren in das neuronale Netz eingegeben werden konnten. Die Merkmale der Röntgenbilder sind Grauwerte im Bereich  $[0, \dots, 255]$ . Die Mehrheit der Bilder der kleinen Datenbank sind im Bezug auf die Helligkeit nicht auf den vollen Grauwertbereich gespreizt. Nur wenige Bilder nutzen den vollen Grauwertbereich aus. Der mittlere Grauwert der Bilder liegt bei 12. Es zeigte sich, dass das Neuronale Netz ein besseres Lernverhalten zeigte, wenn der Arbeitspunkt der Sigmoid-Funktion auf den mittleren Grauwert der Bilder abgebildet wurde. Dazu wurde der Wertebereich der Eingangsbeschriftung des Neuronalen Netzes auf  $[0, 1]$  festgelegt. Aus dem oben genannten Grund wurden die Grauwerte nicht einfach linear auf  $[0, 1]$  abgebildet, sondern durch eine partielle lineare Abbildung, wie in Abbildung 15 gezeigt wird. Dieser Aspekt wurde jedoch in dieser Arbeit weiter nicht untersucht. An dieser Stelle sind möglicherweise weitere

Untersuchungen notwendig, um ein gutes Lernverhalten des neuronalen Netzes zu erhalten.

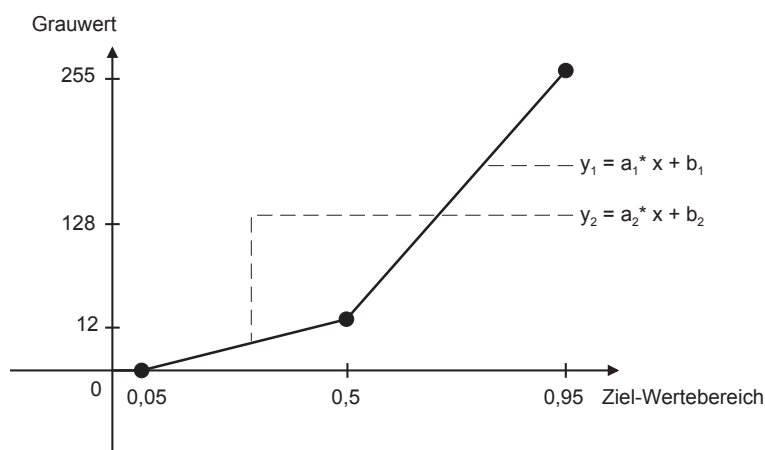


Abbildung 15: Partielle lineare Abbildung des Intervalls  $[0, 1]$  auf die Grauwerte

Ein weiterer wichtiger Punkt in der Trainingsphase eines Neuronalen Netzes, ist das Abbruchkriterium. Auch dieser Parameter muss experimentell bestimmt werden. Ein zu kurzes Training wird dazu führen, dass das Netz nicht gut genug lernt. Wird das Netz zu lange trainiert, dann passt es sich zu gut an die Trainingsdaten an und wird unbekannte Daten schlecht klassifizieren. Als Maß kann hier die Standardabweichung betrachtet werden. Hier kann z.B. ein absoluter Wert vorgegeben werden oder der Mittelwert der letzten 20 Werte betrachtet werden. Eine weitere Möglichkeit ist die Betrachtung der Differenz der letzten Abweichungen. Ist diese kleiner als die Vorgabe, so kann das Training beendet und die Testphase gestartet werden.

## 5 Implementierung

Im Rahmen dieser Studienarbeit wurden die im letzten Abschnitt vorgestellten Methoden implementiert und für die Experimente eingesetzt. Weitere Hilfsprogramme wurden erstellt, die für die Vorverarbeitung der zur Verfügung gestellten Bilder notwendig waren. Hierzu zählt z.B. das Auslesen der Bilder aus Dateien und auch die Extraktion der Merkmale. Die Rohdaten der kleinen Datenbank (82 Trainingsdaten) wurden aus Binärdateien ausgelesen. Die Rohdaten der großen Datenbank (608 Trainingsdaten) waren in ASCII-Dateien gespeichert. Die aus den Rohdaten ausgelesenen Bilder wurden im PGM-Format gespeichert. Die Extraktion der Merkmale wurde dann aus den PGM-Dateien durchgeführt. Die in der Programmiersprache C erstellten Programme werden im Folgenden dokumentiert. Der Quellcode des neuronalen Netzes wurde zur Verfügung gestellt [14]. Es wurde versucht, den Quellcode einheitlich zu gestalten und möglichst gut zu dokumentieren. Dadurch sollten spätere Erweiterungen problemlos durchführbar sein. Die Programme sind sowohl unter dem Betriebssystem Linux als auch unter Windows (Visual C++) ausführbar.

Das Auslesen der Rohdaten, das Speichern des Bildes ins PGM-Format und die Merkmalsextraktion werden mit dem Modul `convert_data_fvector.c` durchgeführt. Die einzelnen Grauwerte werden aus der Quelldatei gelesen und in einer Matrix gespeichert. Nach dem Auslesen wird die Matrix in eine PGM-Datei geschrieben, wobei die Größe der Bilder  $32 \times 32$ -Pixel beträgt. Die Funktion wird mit einem Parameter  $n$  aufgerufen, welcher die Dimension des Merkmalsraumes bestimmt. Der Merkmalsvektor des Bildes wird ermittelt und in eine Zielfeile geschrieben.

Aufruf	<code>convert_data_fvector &lt;filename&gt; &lt;p&gt; &lt;n&gt; &lt;asc_file&gt;</code>	
Optionen	<code>&lt;filename&gt;</code>	Name der Quelldatei, die die Rohdaten enthält
	<code>&lt;p&gt;</code>	p=1 erzeugt filename.pgm Datei, p=0 nicht
	<code>&lt;n&gt;</code>	Dimension des Merkmalsvektors $n^2$ , $1 \leq n \leq 32$
	<code>&lt;asc_file&gt;</code>	Ausgabefile für Merkmalsvektor

Die Prozedur `convert_data_fvector` wurde für jedes Trainingsdatum und jede Dimension  $n^2$  mit  $n \in \{1, \dots, 32\}$  durchgeführt. Dadurch wurden 32 Dateien erzeugt, die die Merkmalsvektoren aller Bilder enthalten. Jede Datei enthält für eine Dimension alle Merkmalsvektoren und die zugehörigen optimalen Grauwerte als Referenzwerte. Die Bezeichnung dieser Dateien geben auch die Dimension der Merkmalsvektoren an: Beispielsweise enthält `data_03.asc` alle Merkmalsvektoren der kleinen Datenbank der Dimension  $3^2 = 9$ .

Die Berechnung des 60%-Wertes eines Bildes wird mittels der Prozeduren `average.c` und `avg_60_608.c` durchgeführt. Zunächst werden mit der Prozedur `avg_60_608.c` die Mengen  $P_i$  bestimmt und die Durchschnittswerte  $p_i$  in einer Datei gesichert. Diese bildet die Eingabe für die Prozedur `average.c`, welche schliesslich den 60%-Wert berechnet und ausgibt.

Bei den folgenden Methoden wurden jeweils eine Trainings- und eine Testversion implementiert. Bei der Trainingsversion wird die Abweichungsrate der Trainingsdaten mittels leaving-one-out ermittelt. Bei der Testversion sind zwei Dateien zu übergeben. Die erste enthält die Trainingsdaten, die zweite die Testdaten. Hierbei werden die Testdaten mittels der Trainingsdaten klassifiziert.

Die Nearest-Neighbour Methode wurde in den Modulen `nn_train.c` und `nn_test.c` implementiert. Analog die Methoden `3nn_train.c` und `3nn_test.c` für 3-Nearest-Neighbour. Dabei bezeichnet `nn_train.c` die Trainings-Version.

Aufruf	<code>nn_train &lt;trainfile&gt; &lt;n&gt; &lt;m&gt; &lt;labels train&gt;</code>	
Optionen	<code>&lt;trainfile&gt;</code>	Merkmalsdatei beliebiger Dimension d (Trainingsdaten)
	<code>&lt;n&gt;</code>	Normalisierung, mit n=1 an, n=0 aus
	<code>&lt;m&gt;</code>	Umkehrung der Normalisierung, mit m=1 an, m=0 aus
	<code>&lt;labels train&gt;</code>	Datei mit Namen der Bilder (Trainingsdaten)

Aufruf	<code>nn_test &lt;trainfile&gt; &lt;testfile&gt; &lt;n&gt; &lt;m&gt; &lt;lab. train&gt; &lt;lab. test&gt;</code>	
Optionen	<code>&lt;trainfile&gt;</code>	Merkmalsdatei beliebiger Dimension d (Trainingsdaten)
	<code>&lt;testfile&gt;</code>	Merkmalsdatei beliebiger Dimension d (Testdaten)
	<code>&lt;n&gt;</code>	Normalisierung, mit n=1 an, n=0 aus
	<code>&lt;m&gt;</code>	Umkehrung der Normalisierung, mit m=1 an, m=0 aus
	<code>&lt;labels train&gt;</code>	Datei mit Namen der Bilder (Trainingsdaten)
	<code>&lt;labels test&gt;</code>	Datei mit Namen der Bilder (Testdaten)

Bei der Implementierung der Kernel-Densities wurde darauf geachtet, dass zur Laufzeit keine numerischen Probleme auftreten. Wenn die Berechnung mit einer Normalisierung der Merkmalsvektoren durchgeführt wird, dann können die Zahlen sehr klein werden. Als Kernfunktion wurde die in Abschnitt 4.3 beschriebene Variante gewählt. Beim Aufruf der Kernel-Densities Methode ist der Parameter  $\alpha$  mit zu übergeben. Die Versionen für Training und Test sind `kd_train.c` und `kd_test.c`.



Aufruf	kd_train <trainfile> <n> <m> <alpha> <labels train>	
Optionen	<trainfile>	Merkmalsdatei beliebiger Dimension d (Trainingsdaten)
	<n>	Normalisierung, mit n=1 an, n=0 aus
	<m>	Umkehrung der Normalisierung, mit m=1 an, m=0 aus
	<alpha>	Wert von $\alpha$
	<labels train>	Datei mit Namen der Bilder (Trainingsdaten)

Aufruf	kd_test <trainfile> <testfile> <n> <m> <alpha> <labels> <labels>	
Optionen	<trainfile>	Merkmalsdatei beliebiger Dimension d (Trainingsdaten)
	<testfile>	Merkmalsdatei beliebiger Dimension d (Testdaten)
	<n>	Normalisierung, mit n=1 an, n=0 aus
	<m>	Umkehrung der Normalisierung, mit m=1 an, m=0 aus
	<alpha>	Wert von $\alpha$
	<labels train>	Datei mit Namen der Bilder (Trainingsdaten)
	<labels test>	Datei mit Namen der Bilder (Testdaten)

Die Module `mlp_module.c` und `mlp_module.h` implementieren die neuronalen Netze und wurden für diese Arbeit zur Verfügung gestellt [14]. Es können beliebige Netze konstruiert und trainiert werden. Das Trainieren wird mittels des im Modul implementierten error-backpropagation Algorithmus durchgeführt. Training und Test sind in den Modulen `mlp_train.c` und `mlp_test.c` implementiert. Einige Parameter sind wegen der verschiedenen Experimente nur im Quellcode veränderbar. Das Training ist folgendermaßen implementiert. Nach jeder Präsentation eines Trainingsmusters wird eine Gewichtsmodifikation durchgeführt, bis alle Muster durchgelaufen sind. Nach jedem Durchlauf des Trainingssets wird die Abbruchbedingung überprüft. Nach dem Training werden dem Netz die unbekanntenen Muster präsentiert und die Fehlerrate wird bestimmt.

Aufruf	mlp_test <trainfile> <testfile> <n> <m> <labels> <labels>	
Optionen	<trainfile>	Merkmalsdatei beliebiger Dimension d (Trainingsdaten)
	<testfile>	Merkmalsdatei beliebiger Dimension d (Testdaten)
	<n>	Normalisierung, mit n=1 an, n=0 aus
	<m>	Umkehrung der Normalisierung, mit m=1 an, m=0 aus
	<labels train>	Datei mit Namen der Bilder (Trainingsdaten)
	<labels test>	Datei mit Namen der Bilder (Testdaten)

## 6 Ergebnisse und Diskussion

In diesem Abschnitt werden die Ergebnisse präsentiert, die bei den Versuchen mit den verschiedenen Methoden erzielt wurden. Als Vergleichswert dient die Vorgabe der 60%-Methode bei der kleinen Datenbank (Tabelle 1). Abbildung 16 zeigt die Anzahl der Abweichungen aufgetragen auf den absoluten Fehler der 60%-Methode. Die Grafik zeigt, dass fast alle Grauwerte überschätzt werden (wg. der Berechnung:  $\hat{y} - y$ ).

Die Grauwertbestimmung mit den Methoden wurde bei den Versuchen auf drei Arten durchgeführt:

1. Ohne Normalisierung.
2. Mit Normalisierung aber ohne Umkehrung.

Standardabweichung	21.87
minimale Abweichung	0
maximale Abweichung	45

Tabelle 1: Ergebnisse der 60%-Methode

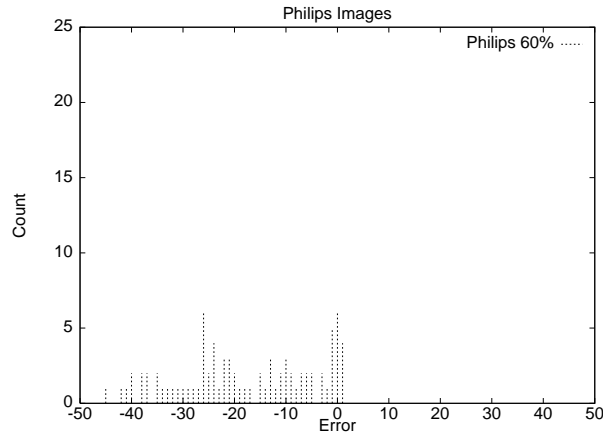


Abbildung 16: Verteilung der absoluten Fehler bei der 60%-Methode

### 3. Mit Normalisierung und mit Umkehrung der Normalisierung.

An dieser Stelle ist zu bemerken, dass die Ergebnisse mit Normalisierung und Umkehrung im Vordergrund stehen, da durch diese Einstellung die ermittelten Grauwerte unabhängig von der Gesamthelligkeit sind. Die Versuche mit den beiden anderen Modi wurden aus Interesse zum Vergleich bei Nearest-Neighbour und Kernel-Densities auch durchgeführt. Die Ergebnisse beziehen sich nur auf das Training, welches durch „leaving-one-out“ durchgeführt wurde.

Zunächst wurde die Nearest-Neighbour Methode untersucht. Tabelle 2 zeigt eine Auswahl der Ergebnisse. Angegeben ist die Dimension des Merkmalsraumes, die Normalisierung, minimale und maximale Abweichung und die Standardabweichung. Das beste Ergebnis wird bei Merkmalsdimension 49 erreicht, wenn Normalisierung und Umkehrung deaktiviert sind. Wenn beide Normalisierungen aktiv sind wird bei Dimension 64 das beste Ergebnis erreicht. Zu diesem Ergebnis wird in Abbildung 17 die Verteilung der absoluten Fehler gezeigt. Abbildung 18 zeigt Ergebnisse der verschiedenen Normalisierungsmodi, wobei die Standardabweichung gegen die Dimension aufgetragen ist. Es fällt auf, dass die Normalisierung bei diesen Experimenten keine positive Wirkung zeigt. Ein Grund ist hier möglicherweise die geringe Datenmenge. Es ist jedoch wichtig, dass aufgrund der Rahmenbedingungen beide Normalisierungen notwendig für einen Einsatz in der Praxis sind.

Dimension	Normal./Umkehrung	maximale Abweichung	Standardabweichung (MSE)
49	- / -	20	4.16
64	- / -	29	4.37
49	+ / -	29	4.74
64	+ / -	29	4.44
49	+ / +	31	5.17
64	+ / +	30	<b>5.15</b>

Tabelle 2: Ergebnisse mit Nearest-Neighbour

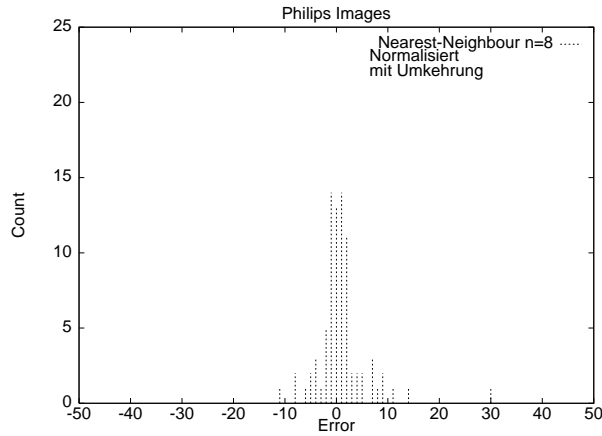


Abbildung 17: Verteilung der absoluten Fehler bei Nearest-Neighbour. Merkmalsdimension ist 64, Normalisierung und Umkehrung sind aktiv

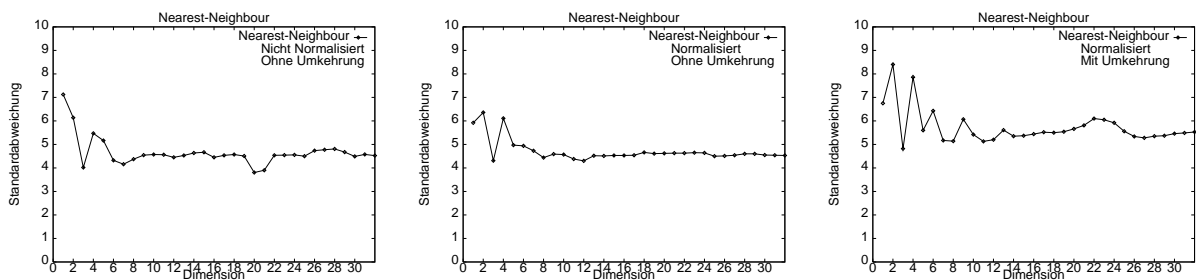


Abbildung 18: Ergebnisse von Nearest-Neighbour. Standardabweichung aufgetragen gegen die Dimension. Ohne Normalisierung (links), mit Normalisierung ohne Umkehrung der Normalisierung (mitte), mit Normalisierung und mit Umkehrung (rechts)

In Tabelle 3 werden die Ergebnisse von 3-Nearest-Neighbour gezeigt. Im Vergleich zum Nearest-Neighbour erkennt man, dass bei Merkmalsdimension 49 eine leichte Verbesserung erreicht wird. Bei 3-Nearest-Neighbour können Ausreißer durch die anderen Nachbarn gedämpft werden und damit das Ergebnis verbessert werden. Auf der anderen Seite können gute Schätzungen des ersten Nachbarn durch die nächsten Nachbarn wieder verschlechtert werden. Im Kontext der Grauwertbestimmung wird dieser Aspekt jedoch nicht so deutlich, da nur eine kleine Datenbank zur Verfügung steht. Die Ergebnisse von Nearest-Neighbour zeigen insgesamt eine deutliche Verbesserung gegenüber der 60%-Methode. (Reduktion der Standardabweichung von 21.9 auf 4.8)

Dimension	Normal./Umkehrung	maximale Abweichung	Standardabweichung (MSE)
49	- / -	26	4.16
64	- / -	28	4.33
49	+ / -	25	3.95
64	+ / -	27	4.45
49	+ / +	26	<b>4.80</b>
64	+ / +	29	5.82

Tabelle 3: Ergebnisse mit 3-Nearest-Neighbour

Bei den Kernel-Densities wurde in mehreren Experimenten der nahezu optimale Wert für  $\alpha$

bestimmt. Tabelle 4 zeigt die Resultate mit den ermittelten  $\alpha$ -Werten. Die Ergebnisse zeigen, dass der Wert von  $\alpha$  nicht den theoretischen Erwartungen entspricht, sondern viel zu klein ist. Die in dieser Studienarbeit durchgeführten Experimente reichen jedoch nicht aus, um diese Tatsache genauer zu analysieren. Die Versuche zeigten, dass die Lösung dahin tendiert, das nur eine geringe Anzahl der Daten zum Ergebnis beitragen sollte. Wird  $\alpha$  sehr klein, so läuft das Verfahren zum Nearest-Neighbour, d.h. nur noch der nächste Nachbar bestimmt das Ergebnis. Auch hier wurde das beste Ergebnis mit Standardabweichung 3.38 erreicht, wenn beide Normalisierungen deaktiviert waren. Durch geeignete Suche wurde das optimale  $\alpha$  für diese Dimension bestimmt und beträgt 0.045. Das beste Ergebnis mit Normalisierung und Umkehrung wird bei Dimension 49 mit Standardabweichung 4.16 erreicht. Ähnlich zu Nearest-Neighbour, sind die Ergebnisse mit Normalisierung der Daten, schlechter als ohne Normalisierung. Insgesamt sind die Ergebnisse von Kernel-Densities besser als Nearest-Neighbour.

Dimension	Normal./Umkehrung	$\alpha$	max. Abweichung	Standardabweichung (MSE)
49	- / -	0.045	22	3.38
64	- / -	0.025	27	3.78
49	+ / -	0.060	27	3.86
64	+ / -	0.072	26	3.93
49	+ / +	0.072	28	<b>4.16</b>
64	+ / +	0.066	28	4.40

Tabelle 4: Ergebnisse von der Kernel-Densities Methode

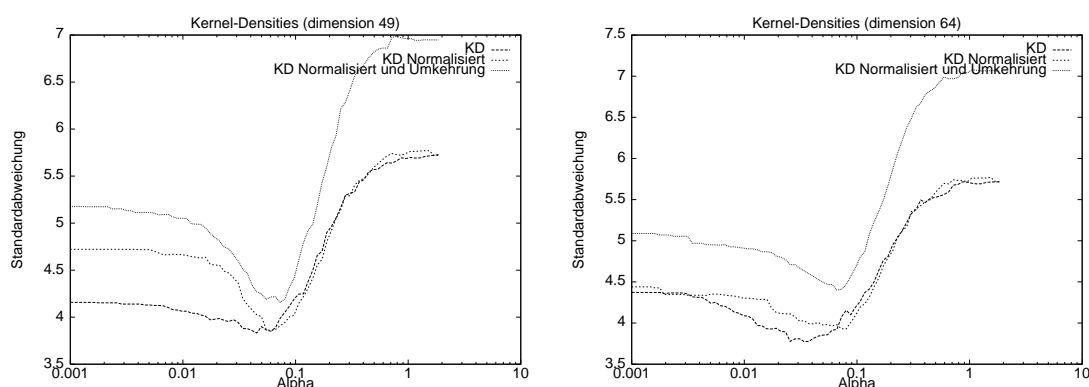


Abbildung 19: Die Bestimmung des optimalen Wertes für  $\alpha$  wurde bei der Methode der Kernel-Densities experimentell durchgeführt. Dabei wurde die Standardabweichung logarithmisch gegen  $\alpha$  bei Dimension 49 (links) und Dimension 64 (rechts), aufgetragen.

Zum Vergleich zu den bisherigen Ergebnissen werden im folgenden die Ergebnisse der neuronalen Netze gezeigt, die im Rahmen dieser Studienarbeit erzielt wurden. Für die Experimente wurde das in Abschnitt 4.4.2 beschriebene neuronale Netz verwendet. Dieses besteht aus einer Eingangsschicht, einer verdeckten Schicht und einer Ausgangsschicht. Die Anzahl der Neuronen in der verdeckten Schicht wurde variabel gelassen, sodass die Versuche mit unterschiedlicher Neuronenzahl durchgeführt werden konnten. Wegen der vielen Tests wurde hier nur noch mit Normalisierung gearbeitet. Wegen der längeren Laufzeiten des neuronalen Netzes sind die Versuche nur auf die Merkmalsdimension 64 beschränkt worden, da bei dieser Größenordnung die Berechnungen stabil waren, wie aus den vorherigen Ergebnissen zu sehen ist (vgl. Abb. 18).

In Tabelle 5 sind die Ergebnisse für die kleine Datenbank aufgelistet. Wie zu erkennen ist, bestand die verdeckte Schicht aus 10 bis 40 Neuronen. Bei den durchgeführten Versuchen, war

das Abbruchkriterium durch den Mittelwert der letzten 20 Standardabweichungen (SMSE, *sliding mean square error*) definiert. Fällt dieser Mittelwert unter eine vorgegebene Schranke, wird das Training beendet und dem Netz das unbekannte Muster präsentiert. Das beste Ergebnis wurde bei 35 Neuronen und einer Schranke  $3.0 \cdot 10^{-4}$  erzielt. Bei dem Vergleich der Ergebnisse bei gleicher Neuronenzahl fällt auf, dass erst eine geeignete Trainingsdauer das bessere Ergebnis liefert. Ein zu kurzes oder zu langes Training kann dazu führen, dass das Netz schlecht generalisiert. Bei 35 Neuronen und Schranke  $5.0 \cdot 10^{-4}$  ist das Netz eventuell noch zu wenig trainiert und bei 35 Neuronen und Schranke  $0.9 \cdot 10^{-4}$  wird das Netz wahrscheinlich übertrainiert. Ein anderer Aspekt ist die zufällige Initialisierung der Gewichte. Je nach Anfangskonfiguration der Gewichte, kann das Netz, bei gleichen Bedingungen, unterschiedliche Ergebnisse liefern. Aufgrund der wenigen Testdurchläufe konnte die optimale Neuronenzahl nicht ermittelt werden. Dabei entsteht jedoch die Gefahr des „Training on the testing data“ [11]. Es ist nur eine Tendenz erkennbar, dass eine geeignete Neuronenzahl im Bereich von etwa 30 Neuronen liegt. Hierzu sind wesentlich mehr Experimente notwendig um eine geeignete Neuronenzahl für die Grauwertbestimmung zu finden. Insgesamt sind die Ergebnisse im erwarteten Bereich und sind ähnlich zu den beiden anderen Methoden. Es fällt auf, dass das beste Ergebnis besser als bei den anderen Methoden ist, allerdings ist die Fehlerrate stark und uneinheitlich von den Parametern abhängig.

Neuronen	Abbruch (SMSE) $\cdot 10^{-4}$	maximale Abweichung	Standardabweichung (MSE)
10	7.0	27	5.58
10	6.0	27	5.01
10	5.0	18	4.95
10	1.0	41	6.50
15	5.0	29	5.40
20	5.0	26	4.81
25	5.0	39	6.04
30	5.0	27	5.16
30	4.5	28	5.50
30	4.0	27	4.75
30	3.0	27	4.62
30	2.0	28	5.43
35	5.0	29	5.67
35	4.0	26	5.08
35	3.5	24	4.38
35	3.0	27	<b>3.98</b>
35	2.5	29	4.92
35	2.0	33	5.87
35	0.9	32	6.45
40	3.0	28	4.33

Tabelle 5: Ergebnisse mit neuronalem Netz, Normalisierung und Umkehrung sind aktiv, Merkmalsdimension ist 64

Die kleine Datenbank mit 82 Bildern stand im Mittelpunkt dieser Studienarbeit. Die große Datenbank mit 608 Bildern war eine hilfreiche Ergänzung. Diese wies jedoch einige Fehler auf. Die Grauwerte von Objekt und Hintergrund waren in nicht wenigen Fällen invertiert. Insgesamt sind die Daten ungleichmässig gewesen, wodurch die guten Ergebnisse der kleinen Datenbank nicht unterstützt werden konnte. Tabelle 6 zeigt die Ergebnisse. Im Vergleich zur 60%-Methode wird hier insgesamt eine Verbesserung um mehr als den Faktor 2 erreicht (30.8 gegenüber 12.7). Wie bei den Ergebnissen der kleinen Datenbank, ist auch hier zu erkennen, dass die Ergebnisse der verschiedenen Methoden ähnlich sind.

Neben einer guten Fehlerrate müssen Methoden für die Grauwertbestimmung Laufzeiten unter 20ms aufweisen [15]. In Tabelle 7 werden die Laufzeiten der Methoden gezeigt. Hierzu

Methode	Einstellung	maximale Abweichung	MSE
60%-Methode (Approx.)	-	81	30.76
Nearest-Neighbour	-	71	14.72
3-Nearest-Neighbour	-	68	13.82
Kernel Densities	$\alpha = 0.025$	109	14.07
Kernel Densities	$\alpha = 0.073$	120	18.71
Kernel Densities	$\alpha = 0.01$	67	<b>12.72</b>
Neuronales Netz	30 Neuronen / Schranke 0.008	67	13.78

Tabelle 6: Ergebnisse bei der großen Datenbank, Normalisierung und Umkehrung sind aktiv, Merkmalsdimension ist 64

wurden die Testläufe bei der kleinen Datenbank bei Merkmalsdimension 64 durchgeführt. Die verdeckte Schicht bei dem neuronalen Netz besteht aus 35 Neuronen. Die Einzellaufzeit ist die wichtigere der beiden Messungen, da im Röntgensystem der Grauwert eines Bildes ermittelt werden muss. Die hohe Gesamtlaufzeit bei dem neuronalen Netz kommt durch das zeitaufwendige Training zustande. Es fällt auf, dass die Einzellaufzeit vom neuronalen Netz im Mikrosekundenbereich liegt. Die Abstandsklassifikatoren haben im Vergleich wesentlich höhere Laufzeiten. Im Realfall würde eine Datenbank wesentlich mehr Trainingsdaten haben, sodass die Einzellaufzeiten bei den Methoden steigen. Hier ist jedoch zu bemerken, dass die Methoden nicht im Bezug auf Geschwindigkeit optimiert worden sind. Durch geeignete Optimierung können die Laufzeiten vermutlich wesentlich verbessert werden. Im praktischen Fall würde man die Methoden als Hardware erstellen wodurch die Geschwindigkeit nochmal deutlich gesteigert würde. Insgesamt erfüllen alle Methoden die Bedingungen der Laufzeit.

Methode	Einzellaufzeit	Gesamtlaufzeit
Nearest-Neighbour	< 1 ms	80 ms
3-Nearest-Neighbour	< 1 ms	40 ms
Kernel-Densities	< 2 ms	160 ms
Neuronales Netz (35 Neuronen)	< 2 $\mu$ s	18.5 min

Tabelle 7: Laufzeiten bei der kleinen Datenbank bei Merkmalsdimension 64, Normalisierung und Umkehrung sind aktiv

Methode	Dimension	maximale Abweichung	MSE
60%-Methode	-	45	21.87
Nearest-Neighbour	64	30	5.15
3-Nearest-Neighbour	49	26	4.80
Kernel-Densities	49	28	4.16
Neuronales Netz	64	28	3.98

Tabelle 8: Gesamtergebnisse der Methoden bei der kleinen Datenbank

In Tabelle 8 sind die Ergebnisse dieser Studienarbeit zusammengefasst. Die Ergebnisse zeigen, dass die untersuchten Methoden durchaus geeignete Verfahren zur Grauwertbestimmung sind. Die Experimente mit den Methoden zeigten, dass die Grauwertbestimmung nahezu unabhängig von der gewählten Methode funktioniert. Es könnte jede der untersuchten Methoden eingesetzt werden. Das Nearest-Neighbour Verfahren bietet den Vorteil der Einfachheit, ist aber eingeschränkt in Bezug auf weitere Optimierungen. Die Methode der Kernel-Densities scheint i.A. besser als der NN-Ansatz zu sein. Das neuronale Netz zeigte auf der kleinen Datenbank das

beste Ergebnis. Neben der 60%-Methode wurde von den Philips Forschungslaboratorien ein weiteres Verfahren für die Grauwertbestimmung untersucht. Dieses basiert auf *clustering*, *shutter detection* und *Segmentierung* und wurde für die Grauwertbestimmung optimiert [15]. Das Verfahren zeigt eine sehr gute Standardabweichung von 1.9. Es ist aber weniger robust und deutlich langsamer in der Laufzeit gegenüber den Methoden die in dieser Arbeit untersucht worden sind.

## 7 Zusammenfassung und Ausblick

In dieser Studienarbeit wurde eine Teilaufgabe eines Röntgendurchleuchtungssystems untersucht. Im Unterschied zur statischen Röntgenaufnahme, ist die Röntgendurchleuchtung eine dynamische Variante einer Röntgenuntersuchung. Ein ganz entscheidender Vorteil dabei ist jedoch, zeitlich ablaufende Vorgänge im Körper sichtbar zu machen. Um die Problemstellung verständlicher zu machen, wurde versucht, das Durchleuchtungssystem in einfacher Form zu beschreiben. Die Problemstellung ist die folgende: Aus der Grauwertverteilung eines digitalen Röntgenbildes ist der Grauwert zu bestimmen, welcher dem mittleren Grauwert des Röntgenobjekts am besten entspricht. Es zeigte sich, dass die Problemstellung den typischen Aufgaben der Mustererkennung entspricht. Formal wird das Problem der Grauwertbestimmung als ein nicht-lineares Regressionsproblem gesehen. Diese Teilaufgabe wurde bisher durch eine sogenannte 60%-Methode durchgeführt, welche jedoch in nicht wenigen Fällen Schwächen zeigt [15]. Daher wurden in dieser Arbeit weitere Methoden untersucht, welche sich für die Grauwertbestimmung eignen. Dabei muss eine Methode neben einer guten Fehlerrate auch bestimmte Laufzeitkriterien erfüllen. Als erster Ansatz wurde das Problem mittels Abstandsklassifikatoren untersucht. Ein weiterer Ansatz zum Vergleich waren neuronale Netze. Die untersuchten Methoden wurden ausführlich beschrieben und die erstellten Programme dokumentiert. Schließlich wurden die vielversprechenden Ergebnisse präsentiert.

Die in dieser Arbeit erzielten Ergebnisse haben aufgrund der zur Verfügung gestellten knappen Datenmenge nur qualitative aber nicht quantitative Aussagekraft. Die Ergebnisse sind jedoch durchaus positiv zu bewerten. Würde eine der Methoden in einem Durchleuchtungssystem verwendet werden, so müsste diese in einem wesentlich größeren Rahmen untersucht werden. Vor allem weil es sich um ein medizinisches Gerät handelt, sind hohe Anforderungen an ein Röntgensystem gestellt. Bei der Methode der Kernel-Densities ist der  $\alpha$ -Parameter genauer zu untersuchen. Bei den neuronalen Netzen sind mehrere Parameter zu untersuchen, um eine gute Netzstruktur zu finden. Die partielle lineare Abbildung der Grauwerte in den Wertebereich  $[0, 1]$  ist möglicherweise keine optimale Lösung. Dies konnte im Rahmen dieser Arbeit nicht genauer untersucht werden. Ein anderer wichtiger Aspekt ist die invariante Objekterkennung. Hier wurde die Invarianz nur im Bezug auf die Helligkeit betrachtet. Die folgenden Arten von invarianter Erkennung sind für das Problem der Grauwertbestimmung zu untersuchen.

- Translationsinvarianz: Unabhängigkeit des Objektes von der Position, d.h. der Verschiebung im Bild.
- Rotationsinvarianz: Unabhängigkeit von der Verdrehung des Objektes im Bild
- Größeninvarianz: Unabhängigkeit von der Größe des Objektes im Bild

## Danksagung

An dieser Stelle möchte ich mich bei Dr. Henning Braess und Dr. Georg Schmitz von den Philips Forschungslaboratorien, Aachen, bedanken, die mir die digitalen Röntgenbilder und viele Informationen zur Verfügung stellten.

## Literatur

- [1] V. Romanovic: Online Dokumentation *Röntgendiagnostik*,  
<http://www.netdokter.de/ratschlaege/untersuchungen/roentgenuntersuchung.htm> (Stand 20.08.2001).
- [2] T. Lehmann, W. Oberschelp, E. Pelikan, R. Repkes: *Bildverarbeitung für die Medizin*. Springer Verlag, Berlin, 1997. S.40 ff.
- [3] C. Caspari: Online Dokumentation *Belichtungsmessung*,  
<http://www.fotolaborinfo.de/foto/fbelicht.htm> (Stand 20.08.2001).
- [4] V. Cherkassky, F. Mulier: *Learning from data, Concepts, theory, and methods*, New York, 1998 - XVIII.
- [5] G.R. Koboltschnig: *Skriptum zur Vorlesung Explorative Datenanalyse*,  
<http://www-stat.uni-klu.ac.at/Lehre/VO-EDA-WS00/eda/node38.html>, Wintersemester 2000/2001.
- [6] H. Huber: Online Dokumentation *Theorie der linearen Regression*,  
<http://www.chemie.unibas.ch/~huber/Statistik/LinReg/LRTheorie/theorie.html>, Intitute of Physical Chemistry, CH-4056 Basel, Switzerland, (Stand 20.08.2001).
- [7] C. Schmid: Online Vorlesungsskript: *Einführung in die Statistische Versuchsplanung mit JMP 4*, [http://www.uni-ulm.de/~cschmid/v2000w/se1/se1\\_2.htm](http://www.uni-ulm.de/~cschmid/v2000w/se1/se1_2.htm), Universität Ulm, Wintersemester 2000/2001.
- [8] D. Keyzers: *Approaches to Invariant Image Object Recognition*, Diplomarbeit am Lehrstuhl für Informatik VI, RWTH-Aachen, Juni 2000.
- [9] M.-O. Gueld: *Inhaltsbasierter Bildzugriff mittels statistischer Objekterkennung* Diplomarbeit am Lehrstuhl für Informatik VI, RWTH-Aachen, Juli 2000.
- [10] W.-M. Lippe: Online Dokumentation Neuronale Netze,  
<http://wwwmath.uni-muenster.de/SoftComputing/lehre/material/wwwnscript/prin.html>, Institut für Informatik, WWU-Münster, (Stand 20.08.2001).
- [11] H. Ney: *Mustererkennung und Neuronale Netze*, Skript zur Vorlesung, RWTH Aachen, Wintersemester 1999/2000.
- [12] H. Ney: *Digitale Signalverarbeitung für Sprache und Bilder*, Skript zur Vorlesung, RWTH Aachen, Sommersemester 2001.
- [13] R. Rojas: *Theorie der Neuronalen Netze*. Eine systematische Einführung, Springer Verlag, Berlin, 1993.
- [14] S. Molau: Quellcode für das Neuronale Netz. Lehrstuhl für Informatik VI, RWTH-Aachen, Juni 2001.
- [15] H. Braess, G. Schmitz, Philips GmbH Forschungslaboratorien, Aachen, persönliche Mitteilung, 2001.