Diplomarbeit zum Thema

„Konzeption und Umsetzung einer RIA zur untersuchungsbegleitenden Erfassung von RNFLT-Scans und Untersuchung von Klassifikatoren für die diagnostische Unterstützung bei neurodegenerativen Erkrankungen am Beispiel der Multiplen Sklerose.“

zur Erlangung des akademischen Grades
Diplom-Informatiker(FH)

vorgelegt dem
Fachbereich Informatik und Medien
der Fachhochschule Brandenburg

Sebastian Bischoff
Matrikelnummer: 20032039
1. November 2009

Betreuer der FH: Dipl. Inform. Ingo Boersch
Betreuer der Firma: Master of Science Sebastian Mansow-Model

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt. Die wörtlich oder sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht.

.....	.....
Ort, Datum	Unterschrift

Danksagung

Mein Dank richtet sich an alle Personen die mich während meines Studiums und vor allem während der Zeit des Schreibens an dieser Arbeit unterstützt haben. Einige Personen sollen hier gesondert aufgeführt werden.
Ich bedanke mich bei meinen beiden Betreuern, Sebastian Mansow-Model und Ingo Boersch für die sehr hilfreiche konstruktive Kritik an der Arbeit sowie die vielen Ideen, ohne die solch ein Ergebnis gar nicht möglich gewesen wäre. Des Weiteren richte ich meinen Dank an Torsten Volland, der mir beim Aufschreiben der mathematischen Formeln sehr hilfreich zur Seite stand. Der nächste Dank richtet sich an meinen Bruder Christian Bischoff, der mich beim einhalten der Form unterstützt hat. Einen riesigen Dank richte ich an Anke Gerlach und Johanna Reihmann, die Korrektur gelesen haben.
Besonderer Dank richtet sich an Alexander Brandt und Markus Bock, die mich mit ihrem medizinischen Wissen während der ganzen Zeit unterstützt haben und mir jede medizinische Frage beantwortet haben. Ebenfalls danke ich den NeuroCure Clinical Research Center der Charité Berlin, für die Bereitstellung der Daten auf denen die Arbeit aufbaut.
Ich möchte auch meinen Eltern und meiner ganzen Familie danken, die mich über die komplette Zeit des Studiums hinweg unterstützt haben. Ein weiterer großer Dank geht natürlich auch an meine Freundin, die es trotz meiner vieler Arbeit mit mir ausgehalten hat.
Abschließend möchte ich noch einmal der gesamten Mediber GmbH danken, dass sie mich so freundlich aufgenommen haben und mir auch bei kleineren Rückschlägen immer den Rücken gestärkt haben.

Inhaltsverzeichnis

1 Zusammenfassung

In dieser Arbeit wurde ein Verfahren zum Erkennen von Multiple Sklerose entwickelt. Hierzu wurden retinale Nervenfaserschichtdickenmessungen eines optischen Kohärenztomographen verwendet. Die Messungen stammen aus einer medizinischen Studie. Zur Erfassung dieser Messungen entstand eine Software, die auch für zukünftige Projekte nutzbar ist. Sie wurde als „Rich-Internet-Applikation“ entwickelt und ist einfach zu erweitern. Mit Hilfe dieser Software gelang es bis jetzt 523 Messungen zu erfassen.
Aus den 256 Einzelmesswerten je Messung wurden durch Verfahren der Bildverarbeitung, Stochastik und Kurvenanalyse 199 neue Attribute generiert. Daran wurde beispielhaft der Einsatz einer „Support Vector Machine“ und des „Naive Bayes“-Algorithmus getestet, um einen Klassifizierer zu finden. Die Ergebnisse zeigen, dass je nach Methode, eine Sensitivität von $39 - 43$ % bei einer Spezifität von 98 % erreicht werden kann. Das ist eine Verbesserung von $34,4$ % zu vergleichbaren bestehenden Verfahren.

2 Aufgabenstellung

Die retinale Nervenfaserschichtdicke (engl.: retinal nerve fiber layer thickness, RNFLT) ist ein moderner Parameter in der Augendiagnostik. Mittels optischer Kohärenztomographie (engl.: optical coherence tomography, kurz OCT) wird hierbei in einem Augen-Laserscan die Faserdicke der Augennerven in 256 radialen Einzelmesswerten bestimmt. Die RNFLT zeigt dabei spezifische Veränderungen in verschiedenen Krankheitsbildern und hält zunehmend Einzug in die Routinediagnostik.
Im Rahmen einer Studie sollen Modelle zum Vergleich der RNFLT-Veränderungen bei Patienten mit unterschiedlichen Erkrankungen gegenüber Kontrollmessungen bei gesunden Probanden bestimmt werden. Bisherige Vergleichsmethoden umfassen lediglich die mittlere Abnahme der Messwerte und berücksichtigen nicht typische Kurvenveränderungen.
Im Rahmen dieser Abschlussarbeit sollen diese Messungen zuerst elektronisch erfasst, verarbeitet und die Auswertungen online verfügbar gemacht werden. Mit Hilfe von Kurvenanalyse und dem Einsatz von Methoden des maschinellen Lernens sollen genauere Lösungen für dieses Klassifikationsproblem entwickelt werden.

3 Theoretischer Teil (Medizin)

3.1 Neurodegenerative Erkrankungen

Neurodegenerative Erkrankungen (NDE) sind chronische Erkrankungen, die durch einen fortschreitenden Untergang von Nervenzellen gekennzeichnet sind. Wichtige Vertreter sind zum Beispiel „Morbus Alzheimer“, „Morbus Parkinson“ und „Multiple Sklerose“. Ihnen gemeinsam ist, dass die Ursachen weitestgehend unbekannt und eine Behandlung schwierig oder nicht möglich ist. Zudem ist durch einen meist schleichenden Beginn die Diagnosestellung schwierig.
NDE treten sehr häufig auf. Sie stellen einen wichtigen Faktor dar, der die Gesundheit, vor allem älterer Menschen, beeinträchtigt. Zudem sind sie durch den demographischen Wandel in Europa ein zunehmender Kostenfaktor im Gesundheitswesen. Allein an Alzheimer erkranken pro Jahr 120.000 Menschen in Deutschland. Vgl.[GMW03]

Multiple Sklerose

Die Multiple Sklerose (MS) ist eine chronische Erkrankung des zentralen Nervensystems (ZNS). Der Begriff „Multiple Sklerose“ steht für mehrfache Verhärtungen, die im zentralen Nervensystem auftreten können. Die Ursache der MS ist unklar. Bekannt ist, dass die MS eine neurodegenerative Erkrankung ist und dass es auch eine immunologische Ursache gibt. In Deutschland sind ca. 120.000 bis 140.000 Menschen daran erkrankt, Frauen sind im Verhältnis 3:2 häufiger betroffen als Männer. Die Erkrankung bricht meist im jungen Erwachsenenalter zwischen dem zwanzigsten und vierzigsten Lebensjahr aus. Die Gefahr an Multiple Sklerose zu erkranken ist regional unterschiedlich, so ist die Verbreitung hauptsächlich bei der weißen Bevölkerung der nördlichen Halbkugel, sowie Australien und des südlichen Afrikas, aber auch verstärkt bei den Farbigen in den Großstädten der USA zu beobachten.
Die Erkrankung ist nicht heilbar, es existieren jedoch Medikamente, die zur Linderung der Symptome beitragen können. Dazu ist aber eine Verlaufsbeobachtung der Erkrankung dringend notwendig. Vgl.[GMW03]

Symptome der Multiple Sklerose

Da die Multiple Sklerose das gesamte zentrale Nervensystem befällt, gibt es sehr viele Symptome. Grundsätzlich kann ein Multiple Sklerose-Patient alle nur erdenklichen neurologischen oder psychiatrischen Symptome entwickeln, einige sind aber häufiger als andere. 40 % der Erkrankten leiden an Kraftlosigkeit, 33 % an Sensibilitätsstörungen, 28 %

weisen eine ein- oder beidseitige Optikusneuritis auf und 18 % sehen Doppelbilder. Ebenfalls leiden viele Patienten an Berührungsschmerzen, Erlöschen der Bauchhautreflexe, Blasenstörungen und Sehstörungen wie Farbsehschwäche, Augenbewegungsschmerzen, eine unscharf begrenzte Papille oder eine temporale Papillenabblassung. Vgl.[GMW03]

Diagnose der Multiple Sklerose

Multiple Sklerose ist eine Erkrankung mit vielen „Gesichtern“. Die Symptome sind vielfältig und werden anfänglich oft übersehen. Sie sind abhängig vom verletzten Hirnareal. Oft dauert es Jahre, bis die Diagnose feststeht. Bis heute gibt es kein eindeutiges Diagnoseverfahren, mit dem die Erkrankung sicher diagnostiziert oder ausgeschlossen werden kann. Die 2001 vorgestellten und 2005 überarbeiteten „McDonald-Kriterien“ legen die Vorgehensweise der Diagnose einer MS fest. Demnach sollte zur genaueren Abklärung einer MS nahezu immer eine Magnetresonanztomographie (MRT) durchgeführt werden. Alle bildgebenden Verfahren werden hiernach als sehr sinnvoll und mit hoher Bedeutung eingeschätzt.
Zur Diagnose der MS gehört aber nicht nur die Erstdiagnose. Da die MS nicht heilbar ist, empfiehlt sich eine lebensbegleitende regelmäßige Untersuchung. Eine solche Verlaufsdiagnose sollte in regelmäßigen Abständen stattfinden. Dazu werden teilweise jährliche Untersuchungen und auch MRT-Messungen durchgeführt. Vgl.[PRE+05]

3.2 Anatomische Grundlagen

3.2.1 Zentrales Nervensystem

Das menschliche Gehirn besteht aus etwa 100 Milliarden Nervenzellen, die von einer Vielzahl weiterer Zellen unterstützt werden. Die Nervenzelle, auch Neuron genannt (siehe Abbildung 3.1), ist darauf spezialisiert, Reize zu übertragen. Sie besteht aus dem Zellkörper und einem Axon. Der Zellkörper besitzt alle Teile, um Reize auszulösen, das Axon ist die Leitung, über die der Reiz weitergeleitet wird. Eine menschliche Nervenzelle kann über einen Meter lang und nur wenige μm breit sein. Das Axon ist von Myelinscheiden umschlossen, die im ZNS von Oligendrozyten gebildet werden und Myelin abscheiden. Das Myelin dient als Isolation, was die Übertragung der Reize um das bis zu Zehnfache beschleunigt. Das Nervensystem teilt sich in das zentrale und das periphere Nervensystem. Zum ZNS zählen das Gehirn, Rückenmark und viele im Kopf befindliche Sinnesorgane. Aus dem Gehirn ragen 12 Hirnnerven heraus, die zum ZNS zählen. Darunter befindet sich auch der Sehnerv. Die Nerven des ZNS grenzen sich deutlich durch ihre zelluläre Zusammensetzung von denen des peripheren Nervensystems ab. Vgl.[GMW03]

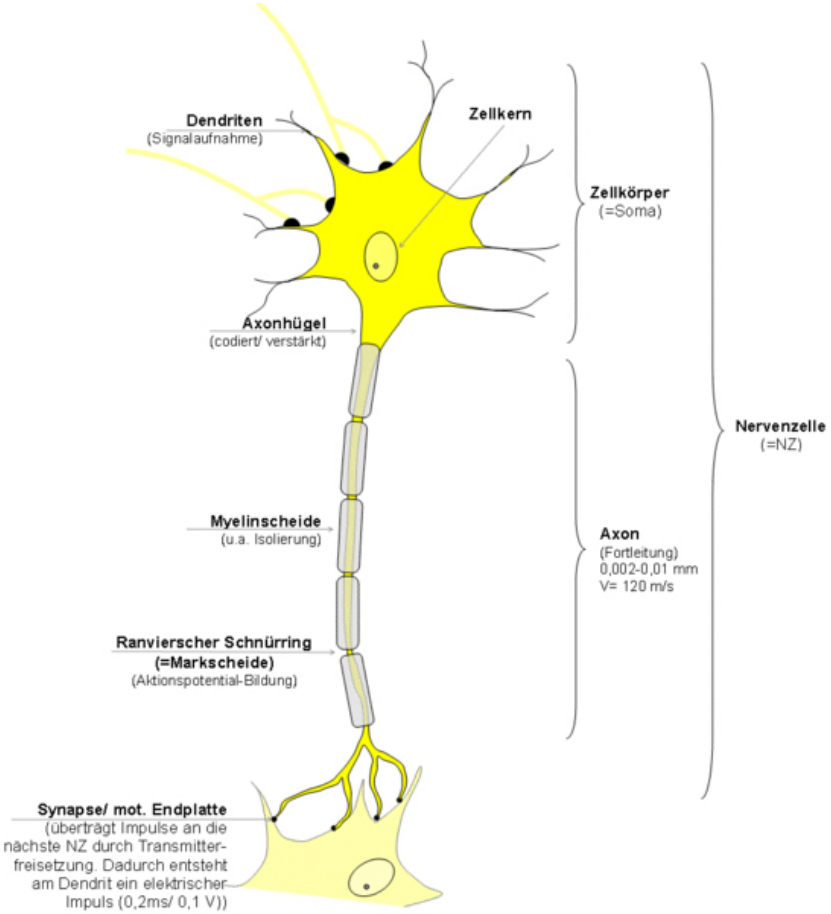


Abbildung 3.1: Schematische Darstellung einer menschlichen Nervenzelle [Hof06]

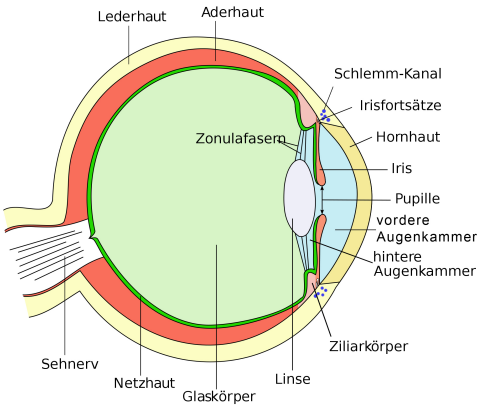


Abbildung 3.2: Schematische Darstellung eines Auges [JG08]

3.2.2 Auge

Der Sehnerv endet im Auge, welches in Abbildung 3.2 schematisch zu sehen ist. Dort geht der Sehnerv in die retinale Nervenfaserschicht (engl.: retinal nerve fiber layer, RNFL), der Retina über. Die Retina oder Netzhaut liegt zwischen dem Glaskörper und der Aderhaut und hat folgende Funktion. Sie detektiert einfallendes Licht und wandelt es in elektrische Impulse um, die dann zum Sehnerv und schließlich zum Gehirn weitergeleitet werden. Die Nervenfasern in der Retina sind für die Diagnose besonders interessant, denn sie zählen zu den Hirnnerven und damit zum ZNS, außerdem sind sie zur Untersuchung besonders gut zu erreichen.

3.3 Optische Kohärenztomographie

Die Optische Kohärenztomographie (engl.: optical coherence tomography, OCT) ist ein nicht invasives, bildgebendes Verfahren zur tiefenaufgelösten Darstellung von biologischem Gewebe mit einer Eindringtiefe von ca. 1-3 mm. Es ähnelt dem bildgebenden Verfahren mit Ultraschall, jedoch wird bei der OCT kein Schall, sondern Licht verwendet. Die OCT verwendet beim Auge energieschwache Laserstrahlen geringer Kohärenzlänge, die mittels Interferometer Aufnahmen mit sehr hoher Auflösung, etwa 20 mal höher als beim Ultraschall, erreichen kann. Vgl.[FCZ+06]
Ein Interferometer spaltet eine Lichtwelle in zwei Wellen auf. Die erste durchläuft einen Referenzweg innerhalb des Messgerätes, die andere den Messweg. Die reflektierten Wellen treffen dann wieder gemeinsam auf den Detektor und erzeugen dort ein Interferenzbild. Von dem Abstand der Wellen im Interferenzbild kann auf die Messwege und somit Abstände im zu messenden Medium geschlossen werden. Die Funktionsweise ist in Abbildung 3.3 bildlich dargestellt. Vgl.[Med03]
Mit diesem Verfahren kann die Dicke der Nervenfaserschicht der Retina in verschiedenen Bereichen abgebildet werden. Das Verfahren wird bei vielen Augenerkrankungen bereits angewandt, zum Beispiel bei einer Lochbildung der Netzhautmitte (Makulaforamen), dem Grüner Star (Glaukom) oder bei einer Netzhautveränderung aufgrund von Diabetes. Neu ist, dass man in Studien versucht dieses Verfahren auch zur Diagnose von Multipler Sklerose zu verwenden.

Messung der Nervenfaserschichtdicke

Die Nervenfaserschichtdicke der Retina (engl.: retinal nerve fiber layer thickness, RNFLT) wird mittels OCT gemessen und in μm angegeben. Für diese Arbeit wurde ein spezieller Scan verwendet, er heißt „fast RNFL Thickness 3.4“. Bei dieser Messung werden die Nervenfasern in einem festen Radius von 3,4 mm um den Sehnerv aufgenommen. Anhand dieser radialen Aufnahme wird die Dicke der retinalen Nervenfaserschicht mit 256 Einzelwerten gemessen. Dabei kann man gut zwischen zellkernreichen und zellkernarmen Schichten unterscheiden, da in beiden Arten von Schichten das Licht sehr unterschiedlich gebrochen wird. Durch das unterschiedliche Brechungsverhalten in den Schichten benötigen einige Frequenzen des Lichtes länger zum Detektor. So können die Abstände

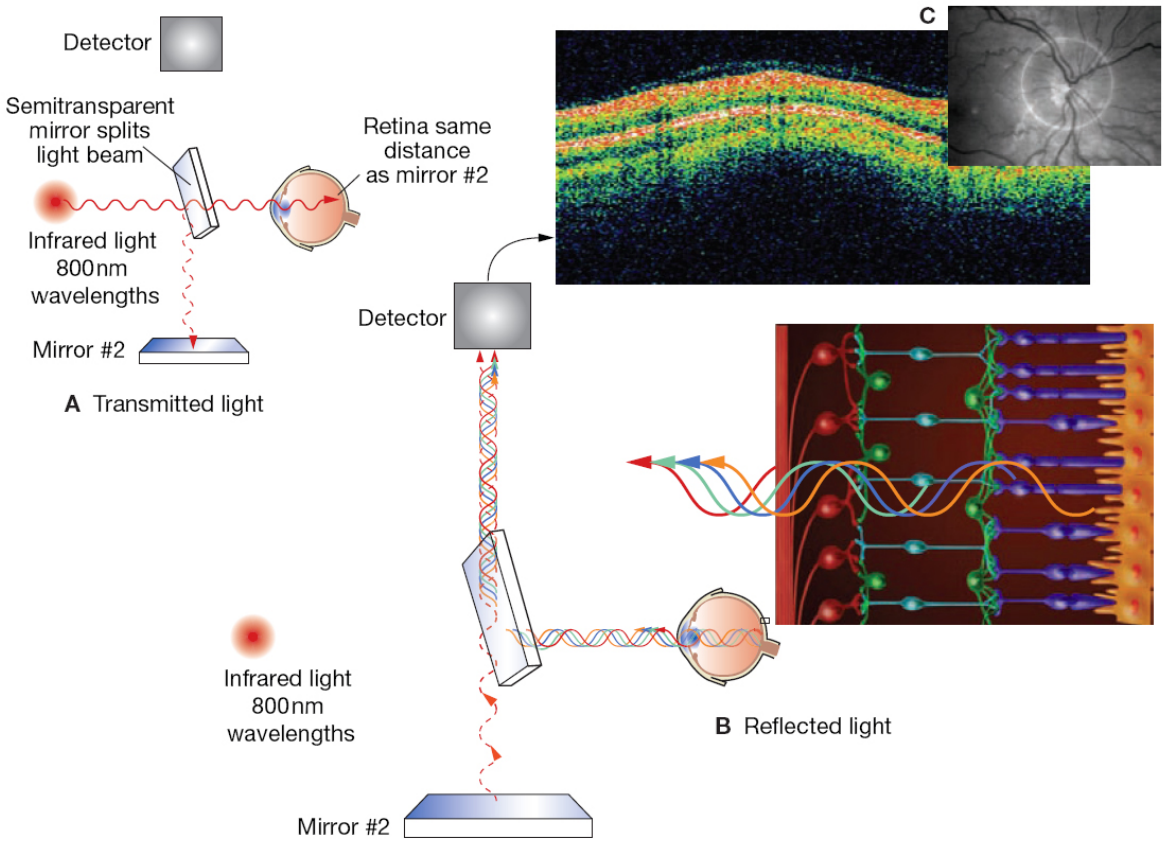


Abbildung 3.3: Schematische Darstellung der Funktionsweise eines OCT Gerätes.
A) Infrarotes Licht geringer Koherenzlänge wird auf einen halbdurchlässigen Spiegel geleitet, ein Teil geht in das Auge, der andere auf einen Spiegel.
B) Jede Wellenlänge des Lichtes kann mehr oder weniger Gewebeschichten durchdringen, bevor es reflektiert wird. Auch das Licht auf dem zweiten Medium (Spiegel) wird reflektiert. Durch die unterschiedlichen Laufzeiten des Lichtes entsteht am Detektor ein Überlagerungsmuster, welches ausgewertet wird.
C) Ein Bild des Augenhintergrundes, aufgenommen vom OCT Gerät, mit dem Messkreis eines „fast RNFL Thickness 3.4“ Scans um den Sehnerv. Links daneben das Ergebnis einer solchen Messung als bildliche Darstellung.
Die Abbildung stammt aus der Veröffentlichung [FFF+08].

detektiert und graphisch dargestellt werden. Die Dicke wird bestimmt, indem die Grenzen der Nervenfaserschichten in das Scanbild eingetragen werden (siehe weiße Linien in Abbildung 3.5). Der Abstand zwischen den beiden Linien ist die Dicke der Nervenfaserschicht.
Das Ergebnis eines solchen RNFLT-Scans ist demzufolge eine Messreihe von 256 Werten der Nervenfaserschichtdicke in μm in einem Kreis von 3,4 mm um den Sehnerv.

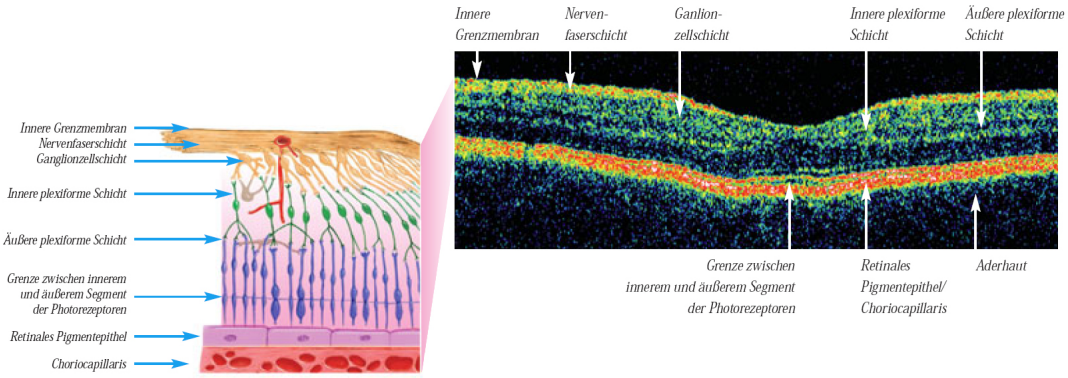


Abbildung 3.4: Grafik mit den Schichten der Retina und dem Bild eines RNFLT Scan, aus [Med03].

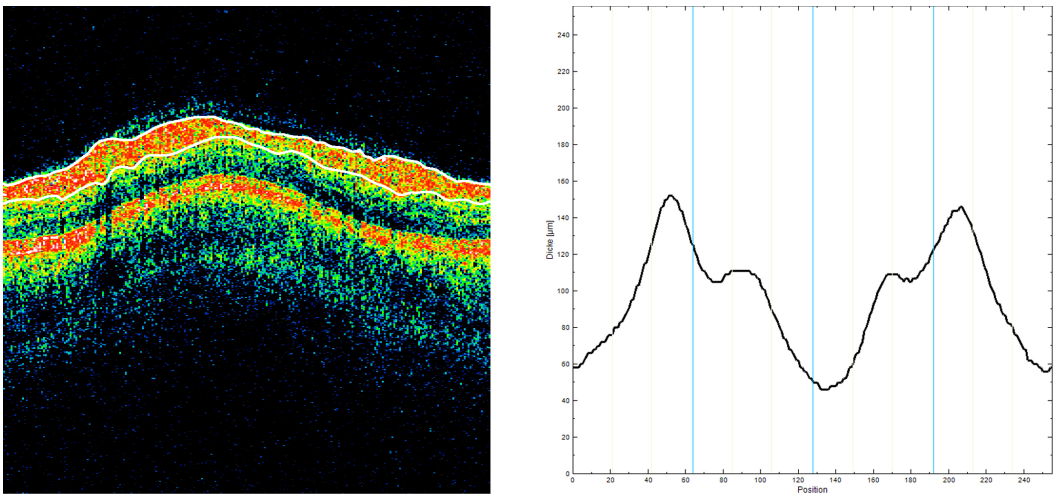


Abbildung 3.5: Bild einer RNFLT Messung, mit eingezeichneten weißen Trennlinien um die Nervenfaserschicht und entsprechender Messreihe, die daraus entsteht.

3.4 Motivation

Die Diagnose der MS ist auch heute noch sehr schwierig. Ein MRT liefert zwar gute Ergebnisse, doch kann man es aus Kostengründen nicht immer einsetzen. Gerade die Kosten einer jährlichen Verlaufskontrolle mit einem MRT bei allen MS Patienten wären viel zu hoch. Ein MRT Gerät kostet zwischen 0,5 Mio. und 1,5 Mio. Euro, nimmt meist ein ganzes Zimmer ein und muss auf supraleitende Temperaturen gekühlt werden. Dies verursacht enorme Kosten. Dazu kommt die demographische Lage in der sich Deutschland befindet, es gibt immer mehr ältere und immer weniger junge Menschen. Wer soll all diese Kosten tragen? Es braucht also ein gutes und gleichzeitig preiswertes Verfahren zur Diagnose der MS, insbesondere für die Verlaufskontrolle.
Die MS zeichnet sich durch das Absterben von Nervenzellen des ZNS und Sehstörungen aus. Ein Ausläufer des ZNS ist der Sehnerv, der direkt in der Retina endet. Der Sehnerv ist der am besten zugängliche Gehirnnerv, weshalb man ihn besonders gut und einfach untersuchen kann. Die Untersuchung des Sehnervs mit einem OCT Gerät wird für die Diagnose bereits heute erfolgreich eingesetzt. Der Zusammenhang zwischen dem Absterben der Nervenzellen im ZNS und der Retina wurde in „Optical coherence tomography: a window into the mechanisms of multiple sclerosis“ von Elliot M. Frohman et al. beschrieben. [FFF+08] Auch in „Optical coherence tomography in multiple sclerosis“ von Elliot Frohman et al. wurde schon darauf verwiesen. [FCZ+06]
Ein OCT Gerät ist preiswert und macht sehr genaue Aufnahmen vom menschlichen Gewebe. Es kostet zwischen 50.000 und 100.000 Euro, nimmt nicht mehr Platz als ein herkömmlicher Röhrenfernseher ein und kann mit einem normalen 230 Volt Anschluss betrieben werden. Im klinischen Bereich werden zur Zeit die OCT Geräte überwiegend in der Augendiagnostik eingesetzt.
Zur Auswertung dieser Untersuchung nutzen die Ärzte zur Zeit nur die Mittel der vom OCT Geträt mitgelieferten Software. Diese bietet aber nur eingeschränkte Funktionen der Auswertung. Hier besteht noch enormes Entwicklungspotenzial. Mathematisch kann man noch viele Algorithmen anwenden und sie von einer Künstlichen Intelligenz (KI) auf Relevanz prüfen lassen.
Ziel sollte es sein, die Diagnose der MS zu verbessern und das Messverfahren mit einem OCT Gerät zur Verlaufsdiagnose der MS nutzbar zu machen. Somit könnten erhebliche Kosteneinsparungen für die Zukunft und eine effizientere Diagnose der MS realisiert werden.

4 Theoretischer Teil (Informatik)

In diesem Teil soll ein Überblick über die Techniken der Informationstechnologie gegeben werden, die für diese Arbeit relevant sind. Dazu zählt der große Bereich der Webanwendungen und der Künstlichen Intelligenz.

4.1 Softwareentwicklung mit Webtechnologien

Heutzutage geht der Trend in der Softwareentwicklung weg von reinen Webentwicklungen und reinen Desktopentwickungen hin zu einer Kombination dieser beiden Bereiche. Dazu gibt es eine noch sehr neue Technologie namens „Rich Internet Application“.

4.1.1 Rich Internet Application

Eine leistungsfähige Internet-Anwendung (engl.: rich internet applikation, RIA) vereinigt Leistungsfähigkeit, Interaktivität und eine intuitive Bedienoberfläche in einer Anwendung, die im Internet verfügbar ist. Eine RIA sollte nicht installiert werden müssen, über Internet-Techniken zugreifbar sein und mit dem Nutzer interagieren. Im Jahr 2002 entstand der Name „Rich Internet Application“. Ziel war es, Internetanwendungen benutzbarer, übersichtlicher, einfacher, intuitiver, komfortabler und eher wie eine Desktopanwendung aussehen zu lassen. Anfänglich genügten einige Flash- und AJAX-Applikationen den Anforderungen einer RIA und durften sich so nennen. Seitdem versuchen viele Anbieter Programmiersprachen und Frameworks zu entwickeln, mit denen man die Vorteile von RIAs ausnutzen kann. Die bekanntesten unter diesen Programmiersprachen und Frameworks sind „Flash“, „AJAX“, „Java-Applets“, „Adobe Flex“ und „Silverlight“. Vgl.[Sch09]
Die RIAs bieten Vor- und Nachteile gegenüber normalen Web- oder Desktopanwendungen, diese sieht man in Tabelle 4.1.

4.1.2 Silverlight

Microsoft Silverlight ist ein browser- und plattformübergreifendes .NET Framework und die Basis für Rich Internet Applications und Rich Media Websites von Microsoft. Silverlight-Anwendungen werden vom Webserver heruntergeladen und auf dem Client typischerweise im Browser ausgeführt. Die Kommunikation der Anwendung mit dem Webserver erfolgt mittels HTTP-GET, REST oder Webservices. Für die Programmierung mittels .NET eignen sich unter anderem die ADO.NET Data Services, die Datenbanken automatisch als Webdienst für einen Silverlight-basierten RIA-Client bereitstel-

Vorteile	**Nachteile**
kein Installationsaufwand	fehlende Plugins auf Clientseite
können begrenzt auch offline funktionieren	Inkompatibilitäten im Browser
Berechnungen der Logik kann clientseitig ablaufen	höhere Ressourcenbelastung des Clientrechners
reduzierte Server- und Netzwerklast durch weniger Kommunikation	längere Downloadzeiten
gegebenenfalls Zugriff auf lokales Dateisystem und Peripherie	neue Sicherheitslücken durch installierte Plugins
können in sicherer Umgebung laufen (Browser-Sandbox)	
Nutzung moderner Benutzersteuerung möglich (Drag & Drop)	

Tabelle 4.1: Vor- und Nachteile einer RIA gegenüber normalen Webentwicklungen oder Desktopanwendungen, vgl.[Sch09]

len können. Silverlight ist hinsichtlich seiner UI-Präsentationsschicht abgeleitet aus der Windows Presentation Foundation. Hauptbestandteil der vektorbasierten Grafikdarstellung und der Gestaltung von Anwendungsoberflächen ist das universelle und textbasierte XML-Format XAML (eXtensible Application Markup Language). Während WPF für die grafische Darstellung und Animationen von Windows-Desktop-Anwendungen entwickelt wurde, ist unter dem Codenamen WPF/E (E für Everywhere) eine webfähige Variante entwickelt worden, die mit einem um Elemente und Funktionen reduzierten XAML ausgestattet ist. Diese wird bei Silverlight verwendet. Die Bereitstellung solcher Software geschieht heute vorzugsweise nicht mehr auf eigenen Servern, sondern auf Plattformen im Internet, die extra dafür geschaffen wurden, dies nennt man „Cloud Computing“. Vgl.[WR08]

4.1.3 Cloud Computing

Cloud Computing (auf deutsch „rechnen in der Wolke“) ist ein Begriff aus der Informationstechnologie (IT). Es bezeichnet die Ablage von Daten, Webseiten, Programmen und vielem mehr in gemieteten Rechenzentren. Die Hardware braucht nicht mehr selber angeschafft zu werden, sondern kann je nach Bedarf gemietet werden. Dies ermöglicht eine sehr schnelle Anpassung der Skalierung von im Internet abgelegter Daten. Man kann in Zeiten von Spitzenauslastungen zusätzliche Ressourcen hinzu mieten und sie bei geringer Last schnell wieder kündigen. Cloud Computing wird als kommende technische Revolution im Internet angesehen und erfreut sich immer größerer Beliebtheit. Die bekanntesten Anbieter zur Zeit sind „Amazon Elastic Compute Cloud“, „Google App Engine“, „Salesforce.com“ und „Microsoft Windows Azure“. Obwohl Cloud Computing viele Vorteile bietet, gibt es auch einige Nachteile. So muss man den Anbietern, de-

nen man seine Daten gibt, auch vertrauen. Geschäftsgeheimnisse lassen sich so nicht zu 100 % bewahren. Des Weiteren ist man von einer funktionierenden Internetverbindung abhängig. Jedoch überwiegen die Vorteile, weshalb mehr und mehr Cloud Anwendungen entstehen und auch von den Anwendern gefordert werden. Vgl.[BKNT09]

4.2 Künstliche Intelligenz

Der Bereich der Künstlichen Intelligenz (KI) rückt in der IT-Branche immer mehr in den Vordergrund. Die KI wird bei Robotern, Datenanalysen, Suchalgorithmen und vielen anderen Bereichen schon eingesetzt. In dieser Arbeit geht es vor allem um das Analysieren von Daten, also um das Finden von Strukturen in Daten und das Bestimmen von Klassifikatoren für neue Datensätze. Vgl.[Ert09]

4.2.1 Data Mining

Data Mining bezeichnet das systematische Anwenden von Methoden des maschinellen Lernens zum Erkennen von strukturellen Mustern oder von implizierten, bislang unbekannten und potenziell nützlichen Informationen aus Daten. Dies wird heutzutage immer mehr in der IT eingesetzt, um intelligente Systeme zu erzeugen. Der Data-Mining-Prozess verläuft dabei immer ähnlich. Es fängt mit der Datenaufnahme, Datenaufbereitung und Datenbereinigung an und geht mit der Generierung von neuen Attributen weiter. Danach kann eine Selektion und/oder Bewertung der Attribute erfolgen. Aus dieser kann dann versucht werden, ein Muster zu erkennen. Dies kann auf sehr unterschiedliche Arten geschehen, mit Hilfe eines „Entscheidungsbaumes“, mit einem „Künstlichen Neuralen Netz“, mit „Support Vector Machines“ oder anderen Lernalgorithmen. Mit dem Muster kann man anschließend versuchen, neue Daten einzuordnen. Am Ende solch eines Prozesses sollte immer die Bewertung des Musters im Bezug auf unbekannte Daten stehen (Evaluierung). Vgl.[WF01]

4.2.2 Entscheidungsbaum

Entscheidungsbäume sind eine spezielle hierarchische Darstellungsform von Entscheidungsregeln. Sie haben einen Einfluss im Data Mining und können dort einfache Muster repräsentieren. Ein großer Vorteil von Entscheidungsbäumen ist, dass sie für den Menschen gut erklärbar und nachvollziehbar sind. Dies erlaubt dem Benutzer, das Ergebnis auszuwerten und Schlüsselattribute zu erkennen. Dies ist vor allem nützlich, wenn die Qualität der Daten nicht bekannt ist.
Ein oft benannter Nachteil der Entscheidungsbäume ist die relativ geringe Klassifikationsgüte in reellwertigen Datenräumen. So schneiden die Bäume aufgrund ihres diskreten Regelwerks bei den meisten Klassifikationsproblemen aus der realen Welt im Vergleich zu anderen Klassifikationstechniken wie beispielsweise Neuronalen Netzen oder Support-Vektor-Maschinen etwas schlechter ab. Das bedeutet, dass durch die Bäume zwar für Menschen leicht nachvollziehbare Regeln erzeugt werden können, diese verständlichen

Regeln aber für Klassifikationsprobleme der realen Welt, statistisch betrachtet, oft nicht die bestmögliche Qualität besitzen. Vgl.[Ert09]

4.2.3 Künstliches Neuronales Netz

Ein Künstliches Neuronales Netz ist ein Begriff aus der IT. Es ist ein stark vereinfachtes Modell eines menschlichen Gehirns. Es besteht aus Neuronen und Synapsen, die in mehreren Schichten mit unterschiedlich vielen Neuronen pro Schicht ausgestattet werden können. Die Synapsen sind die Eingänge der Neuronen, die im Lernschritt gewichtet werden. Die Gewichtung ist ein iterativer Prozess, der nach und nach verbessert wird. Die Neuronen sind Impulsgeber, die mit Hilfe eines Schwellwertes entscheiden, ob sie auslösen und einen Reiz weiterleiten oder diesen unterbinden. Vgl.[RW08]

4.2.4 Support Vector Machines

Bei Support Vector Machines (SVM) handelt es sich um ein rein mathematisches Verfahren der Mustererkennung, das in Computerprogrammen umgesetzt wird. Der Namensteil „machine“ weist dementsprechend nicht auf eine Maschine hin, sondern auf das Herkunftsgebiet der SVMs, das maschinelle Lernen. Eine SVM berechnet die Abstände zwischen den Objekten der verschiedenen Klassen. Dann sucht sie mathematische Funktionen, die genau in die möglichst großen Bereiche zwischen den Klassen passen, um sie damit zu trennen. So kann man die Klassen durch mathematische Funktionen (Vektoren) trennen. Vgl.[SC08]

4.2.5 Genetischer Algorithmus

Genetische Algorithmen (GA) sind Algorithmen, die auch nicht analytisch lösbare Probleme behandeln können, indem sie wiederholt verschiedene „Lösungsvorschläge“ generieren. Dabei verändern und kombinieren sich Attribute und unterziehen sich einer Auswahl, so dass diese Lösungsvorschläge den gestellten Anforderungen immer besser entsprechen. Genauer sind GA heuristische Optimierungsverfahren und gehören zu den evolutionären Algorithmen. Sie werden vor allem für Probleme eingesetzt, für die eine geschlossene Lösung nicht oder nicht effizient berechnet werden kann und stehen in Konkurrenz zu klassischen Suchstrategien wie dem A*-Algorithmus, der Tabu-Suche oder dem Gradientenverfahren.
Die Grundidee genetischer Algorithmen ist, ähnlich der biologischen Evolution, eine Menge (Population) von Lösungskandidaten (Individuen) zufällig zu erzeugen und diejenigen auszuwählen, die einem bestimmten Gütekriterium am Besten entsprechen (Auslese). Deren Eigenschaften (Attributwerte) werden dann leicht verändert (Mutation) und miteinander kombiniert (Rekombination), um eine neue Population von Lösungskandidaten (eine neue Generation) zu erzeugen. Auf die neue Generation wird wiederum die Auslese und Rekombination angewandt. Dieser Ablauf wird mehrmals wiederholt. Am Ende bleiben nur die besten Individuen übrig, die am weitesten angepasst sind. Vgl.[WF01]

5 Ziele der Arbeit

In diesem Kapitel wird die Aufgabenstellung noch verfeinert und die Aufgaben strukturiert aufgelistet. Dadurch lässt sich ein detaillierter Arbeitsablauf erarbeiten. Dieser wird durch zu erreichende Ziele definiert, die nachweislich zu erreichen sind.

5.1 Dateneingabe

Es muss ein Datenbestand zur Analyse aufgebaut werden, dies soll über eine webbasierte Eingabemöglichkeit realisiert werden. Diese Eingabemöglichkeit soll mit Hilfe von „Silverlight“ umgesetzt werden und den Ärzten die Möglichkeit geben, drei RNF - Dateien einzulesen und die Inhalte abzuspeichern. Zusätzlich müssen auch personenbezogene anonymisierte Daten gespeichert werden können. Zur Identifikation der Patienten soll eine Identifikationszeichenkette, die im Krankenhaus schon existiert, verwendet werden. Die Eingabemöglichkeiten sollen später noch erweiterbar sein und die Eingabe soll untersuchungsbegleitend möglich sein.

5.2 Datenbereinigung

Lückenhafte Datensätze sollen vervollständigt oder ausgeschlossen werden. Fehlerhafte Datensätze sollen aufgespürt und korrigiert oder ausgeschlossen werden, dies soll auch für falsche Eingaben gelten. Ziel soll es sein, dass der Nutzer möglichst keine falschen Eingaben vornehmen kann.

5.3 Datenerweiterung

Bei der Datenerweiterung sollen neue Attribute berechnet werden, dies soll deutlich weiter gehen als der jetzige Ansatz, in dem nur die mittlere Abnahme der Messwerte verglichen wird. Die Messwerte sollen mit Mitteln der Informatik sowie der Mathematik untersucht werden.

5.4 Datenauswertung

Bei der Datenauswertung soll mittels systematischer Anwendung von Methoden, ein Muster in den Daten erkannt werden. Dieses soll Rückschlüsse auf die Güte der Attribute oder einen Algorithmus zum Trennen der Datensätze in Klassen hervorbringen.

Dazu soll das Freeware Programm „RapidMiner“ benutzt werden. Wichtig ist auch eine Aussage über die Güte der Ergebnisse. Hierbei besonders die Sensitivität, bei sehr hoher Spezifität.

6 Konzeptioneller Teil

In diesem Abschnitt werden alle Überlegungen, die im Vorfeld zur Umsetzung angestellt wurden, zusammengefasst und strukturiert dargestellt.

6.1 Dateneingabe

Die Aufgabe im Abschnitt der Dateneingabe besteht darin, die Daten vom OCT-Gerät zu exportieren und in einer sinnvollen erweiterbaren Struktur abzulegen. Dazu sollen zusätzlich Patientendaten gespeichert werden.

6.1.1 Export der Messungen

Leider existiert keine offene Schnittstelle, um die Datensätze automatisch zu exportieren. Eine Anfrage bei der Herstellerfirma der Software des OCT Gerätes (Stratus OCT) wurde zurückgewiesen. Deshalb muss das Exportieren der Messungen momentan manuell geschehen. Jeder Datensatz wird per Hand in eine Datei gespeichert. Dies dauert pro Datensatz circa fünf Minuten, denn zu jedem Auge werden drei Messungen auf diese Weise exportiert. Die Dateien sollen in einer eindeutigen Struktur abgelegt werden, dies kann eine hierarchische Verzeichnisstruktur sein. Nach dem Export liegen drei Messungen für jedes linke und rechte Auge vor, sofern beide erfolgreich gemessen werden konnten. Eine Messung liefert eine RNF-Datei und eine Textdatei. In der Textdatei sind Daten zu Messdatum, Geburtsdatum und Seite gespeichert und in der RNF Datei stehen die 256 Einzelmessungen im Dezimalformat mit zwei Nachkommastellen.

6.1.2 Datenhaltung

Die Daten sollen in einer leicht skalierbaren Datenbank abgelegt werden. Dies kann durch eine Microsoft SQL Server Express Datenbank realisiert werden. Dort sollen möglichst alle separierbaren Daten in eigenen Tabellen gespeichert werden, um so jederzeit neue Daten ähnlicher Form gut strukturiert einsortieren zu können. Trigger oder Stored Procedures müssen vermieden werden, um die später mögliche Umstellung auf „SQL Data Service“ von Windows Azure zu realisieren. In der Patiententabelle muss die auch im Krankenhaus geführte „ScreeningID“ als Primärschlüssel verwendet werden.
Möglich wäre auch die sofortige Verwendung von „SQL Data Service“ zur Speicherung der Daten. Dies macht jedoch nur Sinn, wenn die komplette Anwendung online als Cloud-Anwendung installiert wird.

6.1.3 Eingabeformen

Zur Eingabe können mehrere Formen sinnvoll sein, eine manuelle Eingabe eines kompletten Datensatzes oder die automatische Eingabe aller elektronisch lesbar vorliegenden Daten als minimaler Datensatz. Die manuelle Eingabe sollte einfach und übersichtlich sein, sie darf wenig Möglichkeiten zur Eingabe falscher Daten bieten und untersuchungsbegleitend durchführbar sein. Realisierbar wäre eine RIA mit einem flachen Oberflächendesign, so kann man schnell auf alle Daten zugreifen und die Software von verschiedenen Standorten sehr gut untersuchungsbegleitend nutzen. Zur Umsetzung wäre „Microsoft Silverlight“ als Cloud Anwendung auf der „Microsoft Windows Azure“ Plattform mit der „Windows LiveID Authentifikation“ zur Anmeldung sinnvoll.
Die Silverlight-Anwendung eignet sich auch zur lokalen Installation auf einem Laptop, so kann man schneller entwickeln und erfüllt leichter die Sicherheitskriterien, die an die Datenübermittlung, Datenspeicherung und Authentifizierung gestellt werden.
Als automatische Eingabe könnte man sich eine Batcheingabe vorstellen, die alle neuen exportierten Messungen selbstständig in die Datenbank einträgt. Dies könnte besonders als Ersteingabe und zur Überprüfung auf fehlende Eingaben genutzt werden. Die Umsetzung könnte als „Windows Forms Applikation“ stattfinden und dem Benutzer unbedingt eine Rückmeldung über Erfolg oder Misserfolg des Imports geben.

Webanwendung ja/nein

Sollte die Eingabesoftware als webfähige Anwendung entwickelt werden? Um diese grundlegende Frage zu beantworten, werden zuerst die Vor- und Nachteile einer Webanwendung in Tabelle 6.1 gegenüber gestellt.

Vorteile	**Nachteile**
kein Installationsaufwand	Datenschutz
schnellere unproblematische Wartung möglich	sichere Authentifizierung der Benutzer nötig
bessere Zugreifbarkeit	längere Entwicklungszeit
zukunftsweisende Technologie	nicht Offline fähig
Plattformunabhänigkeit	

Tabelle 6.1: Vor- und Nachteile der Eingabe als Webapplikation

Zusätzlich zu den aufgelisteten Nachteilen existieren noch Fakten, die aus dem Projekt heraus zu betrachten sind. Innerhalb der Projektplanug steht nur begrenzt Zeit für die Entwicklung zur Verfügung. Hinzu kommt die Tatsache, dass zur Zeit nur eine örtliche Stelle existiert, an der neue Messungen aufgenommen werden.
Aus diesen Gründen ist es sinnvoll, den Entwicklungsaufwand gering zu halten, die Option auf eine reine Webanwendung aber für eine spätere Weiterentwicklung offen zu lassen. Die Software sollte also mit Silverlight umgesetzt werden, jedoch noch nicht als Cloud-Anwendung.

Dadurch bleibt die Option einer Cloud-Anwendung bestehen und die Entwicklung der Eingabesoftware sollte schneller voran schreiten. Zusätzlich ist eine optimale Lösung gefunden, die sich einerseits an den Zeitplan des Projektes hält und andererseits eine zukunftsorientierte Weiterentwicklung ermöglicht.

6.2 Datenbereinigung

Die Datenbereinigung dient dem Zweck, dass keine fehlerhaften oder unvollständigen Datensätze dem DataMining zugeführt werden. Dazu können mehrere Ansätze verfolgt werden. Man kann die fehlerhaften und unvollständigen Datensätze weglassen, doch dadurch würden viele Datensätze verloren gehen, dies soll aber vermieden werden. Eine weitere Möglichkeit ist, die Dateneingabe so zu gestalten und zu beeinflussen, dass möglichst wenige fehlerhafte oder unvollständige Datensätze entstehen. Der zweite Ansatz ist für diese Arbeit favorisiert.
Eine manuelle Eingabemaske soll am Besten nur aus Auswahlmöglichkeiten und nur an dringend benötigten Stellen aus Freitexteingaben bestehen. Ebenfalls soll versucht werden, die Fehlerquelle Mensch noch weiter zu minimieren, indem auf eine manuelle Eingabe größtenteils verzichtet wird. Ein Dateneingabescript soll alle elektronisch zur Verfügung stehenden Daten automatisch erfassen und dabei gegebenenfalls noch Redundanzen ausnutzen um Fehler zu erkennen.
Schließlich sollte auch noch eine manuelle stichprobenartige Betrachtung der eingegebenen Daten geschehen, um eventuell auftretende Unstimmigkeiten in den Daten zu erkennen. Dies kann durch die eingebende Person oder den Data Miner geschehen.

6.3 Datenerweiterung

Bei der Datenerweiterung wird versucht, aus den bereits vorhandenen Werten durch Mutation, Kombination oder Anwendung von Wissen zusätzliche Werte entstehen zu lassen. Durch verschiedenste Fähigkeiten des Menschen, wie Intuition oder das Anwenden von Wissen, sind wir in der Lage, Zusammenhänge durch einfaches Betrachten schneller zu erkennen als ein Computer.
Aus den vorliegenden Werten war deshalb eines sofort zu erkennen, der Verlauf der Messreihe ist, bis auf kleine Ausnahmen, mit einer Kurve zu vergleichen, deren Anfang gleich dem Ende ist. Im Bereich der Mathematik bietet sich zur Analyse solcher Kurven das Verfahren der Kurvendiskussion an.
Ein weiterer Ansatz wäre die Messreihe in Abschnitte zu unterteilen. Der erste Schritt hierbei wäre die Unterteilung in vier Abschnitte (Quadranten). Aus der Medizin heraus ergeben sich für die einzelnen Quadranten bereits anatomische Zusammenhänge. Hierbei können den Quadranten Bereiche des Kopfes zugeordnet werden. Für die Quadranten ist bereits bekannt, dass in manchen die Nervenfaserschichtdicke höher ist als in anderen. Durch diesen anatomischen Hintergrund verspricht die Methode besonders gute Ergebnisse zu liefern. Eine zusätzliche Unterteilung der Quadranten in weitere drei Abschnitte

ist in der Medizin, als Stunden (engl.: hours) bereits bekannt.
Aus dem Bereich der Informatik heraus ist die Bildverarbeitung bekannt. Die Verfahren der Bildverarbeitung kann man gut auf solche Kurven anwenden. Durch Verfahren der Kantendetektion können diese hervorgehoben und anschließend analysiert werden. Eine mögliche Mustererkennung kann auch zielführend sein.
Die Fourier Transformation wäre dafür auch ein sehr guter Ansatz. Da die Messreihe unterschiedliche Schwingungen aufweist, könnte die Analyse dieser Schwingungen interessante und wertvolle Ergebnisse liefern.
Ein letzter Ansatz könnte sein, die Messreihe als Verteilung von zufälligen Ereignissen zu betrachten. Dazu gibt es keinen ersichtlichen Hintergrund, aber es ist eine vollkommen andere Herangehensweise, die zu Ergebnissen führt, die weiter verarbeitet werden können. Wie sinnvoll diese sein werden, kann man nicht abschätzen, aber es ist ein weiterer möglicher Ansatz.

6.4 Datenauswertung

Beim Auswerten der Daten kann man sehr viele Ansätze verfolgen. Drei grundlegende Schritte sollten dabei immer eingehalten werden. In einem ersten Schritt wird ein Setup gesucht, nach dem das DataMining ablaufen soll. Dieses Setup ist der zeitliche Ablauf der einzelnen DataMining Abschnitte. Es ist in zwei entscheidende Teile unterteilt, die Lernalgorithmen und die Bewertungsalgorithmen. Die Lernalgorithmen erstellen ein Modell, mit dem neue Datensätze klassifiziert werden können. Die Bewertungsalgorithmen bestimmen, wie gut diese Modelle funktionieren.

Abbildung 6.1: Aufteilung der Datensätze: Die Lerndaten werden zum Lernen und Optimieren des Modells verwendet. Die Lerndaten bestehen aus 60 % bis 90 % der gesamten Daten. Die restlichen Datensätze zählen zu den Testdaten, die zum Testen des Modells benutzt werden.

6.4.1 Erstellung eines Setup

Als Grundlage für ein gutes Setup sollten mindestens zwei unabhängige Mengen von Datensätzen vorhanden sein, in denen alle Klassen repräsentativ vertreten sind, siehe Abbildung 6.1. Bei der Verwendung von mehr als zwei Mengen treten meist keine Verbesserungen der Klassifikatoren auf, dadurch verbessert sich nur die Aussage über die Qualität des Klassifikators. Festzuhalten ist, dass bei der Verwendung von zu vielen Teilmengen oder einem falschen Verhältnis, die Teilmengen so klein werden können, dass sie die einzelnen Klassen nicht mehr repräsentativ beinhalten. Somit wird sich in dieser Arbeit auf zwei unabhänige Mengen beschränkt.
Ein Setup kann nun wie in Abbildung 6.2 aussehen. Mit der ersten, größeren Menge

(Lerndaten) soll ein optimales Modell trainiert werden. Dazu verändert man so lange die Parameter, bis sich die Performance nicht mehr verbessert. Die Lerndaten werden dazu noch einmal in zwei Teile geteilt. Den Ersten nutzt man zum Lernen und den Zweiten zum Testen. Statt einer Zweiteilung der Daten ist auch eine X-Validierung denkbar. Ist das Ergebnis zufriedenstellend, so nutzt man alle Lerndaten und lernt noch einmal ein Modell mit den gleichen Einstellungen an.
Die bisherigen Bewertungen sind jedoch noch nicht repräsentativ, denn die Ergebnisse entstanden durch Anpassen der Einstellungen, so dass dieses Modell für diese Daten nun gut funktioniert.
Nun stellt sich die Frage, wie dieses Modell auf völlig unbekannte Daten reagieren wird. Um dies zu testen, kann man noch einen Schritt weiter gehen. Dazu hat man die zweite, kleinere Menge vom Anfang noch übrig. Diese Menge hatte noch keinen Kontakt mit dem Modell. Wendet man das Modell nun auf diese unbekannten Daten an und bewertet das Modell erneut, so bekommt man einen zweiten Wert, der zeigt, wie gut das Modell auf unbekannten Daten funktioniert.
Dies soll das grundlegende Setup werden. Hier können nun an verschiedenen Stellen noch zusätzliche Algorithmen eingebaut werden oder an den Parametern der Algorithmen gearbeitet werden. Möglich wäre auch eine Selektion der Attribute, verschiedene Formen der inneren Aufsplittung der Datensätze, andere Lernalgorithmen und unterschiedliche Bewertungsalgorithmen.

6.4.2 Lernalgorithmen

Die Lernalgorithmen sollen ein Modell erstellen, welches die Datensätze den zwei Klassen (gesund/erkrankt) zuordnen kann. Dies kann durch verschiedene Methoden geschehen. Einige Methoden sind künstliche neurale Netze, Entscheidungsbäume, den Nearest Neighbor Algorithmus, Naive Bayes Algorithmus, eine Support Vector Machine oder eine Kombination oder Erweiterung dieser. Eine Erweiterung könnte zum Beispiel die zusätzliche Verwendung eines evolutionären oder genetischen Algorithmus sein. Einige dieser Ansätze sind eher einfacher Natur. Der Nearest Neighbor Algorithmus zum Beispiel merkt sich alle Lerndaten und sucht bei einem neuen unbekannten Datensatz nur den ähnlichsten der gelernten Datensätze, um den neuen zu klassifizieren. Andere haben da weitergehende Ansätze, trotzdem kann es sein, dass die Daten sich von dem einfachen Nearest Neighbor Algorithmus besser als von allen anderen Klassifikatoren zuweisen lassen. Dies hängt stark von der Struktur der Daten ab. Von der Art der Daten hängt auch ab, welche Algorithmen sich überhaupt eignen, denn einige sind besonders für Zahlenwerte und andere für nominale Werte geeignet. Man muss also testen, welcher Ansatz für die Daten die besten Ergebnisse liefert. Dazu sollten mehrere Ansätze getestet und die Ergebnisse bewertet und protokolliert werden, um daraus den bestmöglichen Algorithmus auswählen zu können.

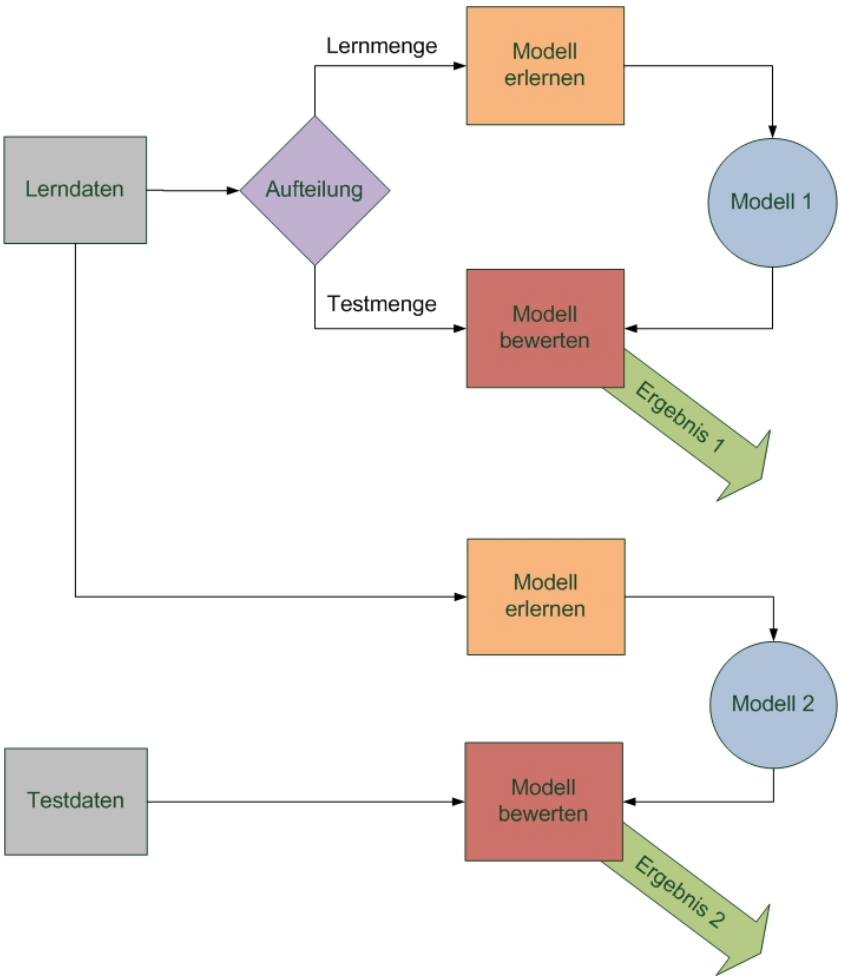


Abbildung 6.2: Geplantes grobes Setup des DataMinings. Die Daten sollen in dieser oder einer ähnlichen Form verarbeitet werden und somit zu zwei Ergebnissen führen, die miteinander vergleichbar sind.

6.4.3 Bewertungsalgorithmen

Die Bewertungsalgorithmen sollen die erlernten Modelle bewerten. Dies geschieht meist durch statistische Auswertung der Ergebnisse. Dazu kann die Standardabweichung, Varianz, Genauigkeit, Trefferquote, der absolute Fehler, die Anzahl der richtig positiv erkannten Elemente und noch vieles mehr berechnet werden. Da schon Untersuchungen zum Mittelwert der Messreihe vorliegen und dort die Sensitivität bei möglichst hoher Spezifität untersucht wurde, sollte dies auch bei diesen Daten untersucht werden.
Gesucht wird also das Modell, welches eine möglichst hohe Sensitivität, bei einer gegebenen Spezifität von 95 % bis 98 % hat. Dies kann man aus der ROC-Kurve (engl.: receiver operating characteristic, ROC) entnehmen, weshalb diese Kurve auch zur Bewertung mit herhangezogen werden kann. Ein guter Wert, welcher auf der ROC-Kurve beruht, ist die Fläche unter der Kurve (engl.: area under curve, AUC), vielleicht kann auch dieser Wert genutzt werden, um die Modelle zu bewerten.

7 Ergebnisse / Implementierung

In diesem Kapitel wird erläutert, wie und mit welchen Mitteln die Umsetzung erfolgt ist. Nicht alle Überlegungen konnten umgesetzt werden. Dies lag zum Teil daran, dass sich auch nach ausführlichen Tests kein erfolgversprechendes Ergebnis zeigte. Andere Überlegungen wurden aufgrund des Projektdrucks zeitlich zurückgestellt. Spezielle Überlegungen wurden weiter verfeinert und werden im Folgenden dargestellt.

7.1 Dateneingabe

Die Dateneingabe gliedert sich in folgende Teile:

- den Export der Messungen vom OCT-Gerät
- die Bereitstellung einer Datenbank
- einem Programm zur manuellen Eingabe eines kompletten Datensatzes
- einem Programm zur automatischen Erfassung neuer Messungen

7.1.1 Export der Messungen

Der händische Export wurde von den Mitarbeitern des „NeuroCure Clinical Research Center“ kurz NCRC der Charité Berlin und mir durchgeführt. Es sind insgesamt 523 Messreihen, davon 306 von gesunden Probanden und 217 von an MS erkrankten Probanden. Die Daten stammen alle von einem Messgerät des Typs „Stratus OCT“ der Firma „Carl Zeiss Meditec AG“. Der Export fand von Anfang November 2007 bis zum 26. September 2009 statt. Diese Messreihen dienen als Grundlage für die Auswertung. Die Eingabe ist noch nicht abgeschlossen, doch wurde der Datenbestand an diesem Punkt eingefroren, um die Analyse durchführen zu können.
Die exportierten Dateien wurden in einer speziellen Dateistruktur abgelegt (siehe Abbildung 7.1). Als oberste Hirarchiestufe dient ein Kürzel für die Klasse, zu der die Messung gehört (HC = healthy control; MS = multiple sklerosis). Die nächste Stufe ist die Identifikationszeichenkette aus vierstelliger Zahl plus Initialen. Die letzte Hierarchiestufe ist der Name der Visite, der aus einem großen „V“ und der Anzahl der Monate seit der ersten Untersuchung besteht. In diesem Ordner werden alle Dateien zu dieser Messung abgelegt.
Auch die Dateien sind in einer klaren Struktur immer gleich benannt. Die Struktur dafür ist:

[Messungsart]_[Pat.ID]_[Visit.ID]_[OD/OS]_[Scan Nr]

Dabei steht „Messungsart“ für die Art der erfolgten Messung. Welche Kürzel dies sein können, kann man in Tabelle 7.1 sehen. „Pat.ID“ für die Identifikationszeichenkette des Patienten, die aus 4 Zahlen und den Initialen besteht. „Visit.ID“ steht für den Namen der Visite, der gleich dem Ordner ist. „OD“ oder „OS“ steht für die Seite, die untersucht wurde, „OD“ ist dabei rechts und „OS“ steht für links. Schließlich folgt noch die „Scan Nr“. Da drei Scans vorliegen, ist dies also eine Zahl zwischen eins und drei. Alle Teile sind durch einen Unterstrich voneinander getrennt. Diese Art der Ablage der Dateien wurde von „Alexander Brandt“ und mir entwickelt und in [BB09] niedergeschrieben.

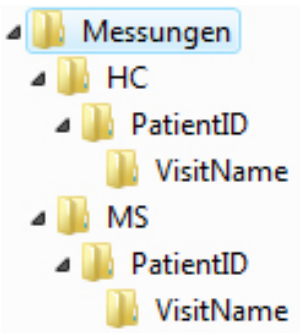


Abbildung 7.1: Verzeichnisstruktur der exportierten Dateien

Kürzel	**Messungsart**
RT	Fast RNFL Thickness 3.4
FMT	Fast Macular Thickness Map
FOD	Fast Optic Disc
RM	Fast RNFL Map
MC	Max Circle

Tabelle 7.1: Arten der möglichen Messungskürzel für die Dateinamen der Messungen

7.1.2 Datenhaltung

Die Daten sollten aber zur weiteren Bearbeitung auch in einer Datenbank zusammen mit den personenbezogenen Daten sinnvoll archiviert werden. Dazu entstand die in Abbildung 7.2 zu sehende Datenbank. Sie sollte skalierbar und der Zugang sicher sein. Die Datenbank wurde mit Microsoft SQL Server Management Studio in der Version 10 erstellt und auf einem Microsoft SQL Express 2008 Server lokal auf einem Laptop installiert.

Struktur der Datenbank

Die Datenbank kann Patienten mit den dazugehörigen Visiten speichern. Zu jeder Visite können zwei Augen mit den entsprechenden Messungen abgelegt werden. Die Struktur der Datenbank sieht man in Abbildung 7.2. Eine Erklärung der Struktur kann man der Tabelle 7.2 entnehmen. Einige Spalten hätten auch in andere Tabellen integriert werden können, doch sollte die Datenbank sehr übersichtlich und einfach zu erweitern sein. Dadurch sind die Tabellen Data_edss, Data_vep und Data_rnflt entstanden.

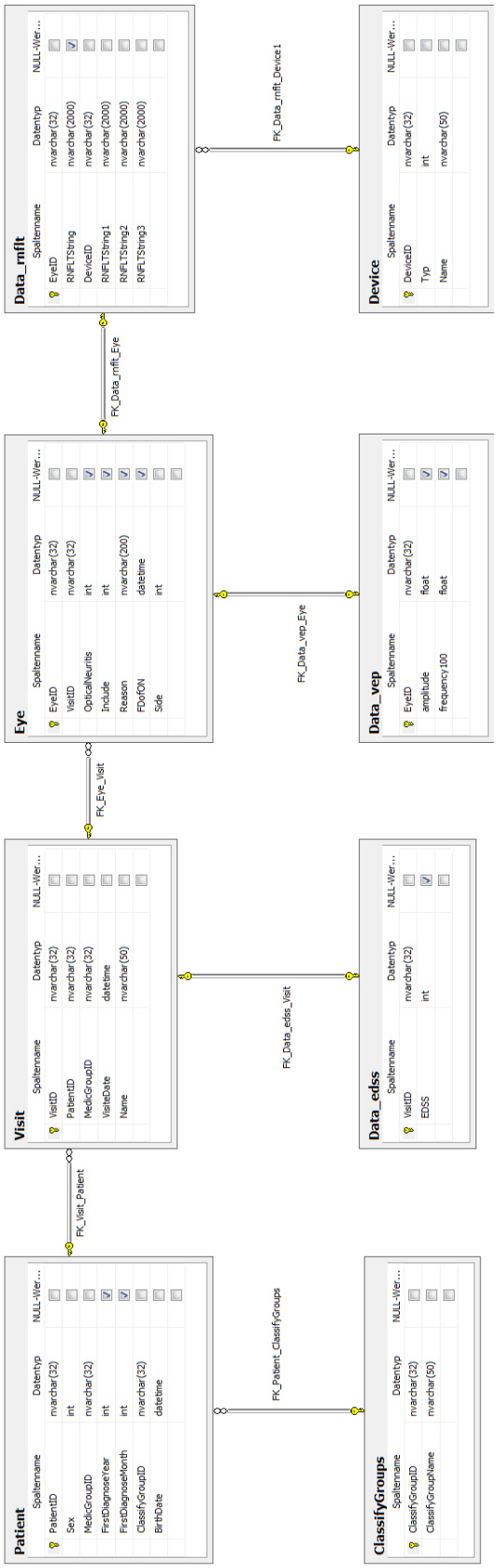


Abbildung 7.2: Diagramm der Datenbankstruktur, erstellt mit Microsoft SQL Server Management Studio

Tabelle	**Spalte**	**Beschreibung**
Patient	PatientID	Identifikationsstring des Patienten, der als Primärschlüssel dient und als Fremdschlüssel für die Datenhaltung des Arztes dient.
	Sex	Geschlecht des Patienten
	MedicGroupID	Identifikationsstring des Arztes des Patienten
	FirstDiagnoseYear	Jahr der Erstdiagnose
	FirstDiagnoseMonth	Monat der Erstdiagnose
	ClassifyGroupID	Identifikationsstring der Klasse des Patienten, also ob und welche Krankheit er hat. (Fremdschlüssel)
	BirthDate	Geburtsdatum des Patienten
Visit	VisitID	Identifikationsstring der Visite, der als Primärschlüssel dient
	PatientID	Fremdschlüssel zur Patient Tabelle
	MedicGroupID	Identifikationsstring des Arztes, der die Visite aufgenommen hat
	VisiteDate	Datum der Visite
	Name	Name der Visite
Eye	EyeID	Identifikationsstring des Auges, der als Primärschlüssel dient
	VisitID	Fremdschlüssel der Visite
	OpticalNeuritis	Wert für das Vorhandensein einer Opticusneuritis
	Include	Wert für den Ausschluss des Auges
	Reason	Grund des Ausschlusses des Auges
	FDofON	Datum der Erstdiagnose der Opticusneuritis
	Side	Seite des Auges
Data_rnflt	EyeID	Identifikationsstring der RNFLT Messung, der eine 1:1 Beziehung mit dem Auge realisiert
	RNFLTString	arithmetisches Mittel der ausgelesenen Messungen
	DeviceID	Fremdschlüssel zur Device Tabelle
	RNFLTString1	erste ausgelesene Messung
	RNFLTString2	zweite ausgelesene Messung
	RNFLTString3	dritte ausgelesene Messung
Device	DeviceID	Identifikationsstring des Gerätes, mit dem die Messung durchgeführt wurde und der als Primärschlüssel dient
	Typ	Typ des Messgerätes
	Name	Name des Messgerätes
Data_vep	EyeID	Identifikationsstring der VEP Untersuchung, der eine 1:1 Beziehung mit dem Auge realisiert
	amplitude	Amplitudenwert der VEP Untersuchung
	frequency100	Frquenzwert der VEP Untersuchung
Data_edss	VisitID	Identifikationsstring des EDSS Untersuchung, der eine 1:1 Beziehung mit der Visite realisiert
	EDSS	Wert der EDSS Untersuchung
ClassifyGroups	ClassifyGroupID	Identifikationsstring des Klasse, der als Primärschlüssel dient
	ClassifyGroupName	Bezeichnung der Klasse

Tabelle 7.2: Erster Teil der Beschreibung der Spalten der Datenbankstruktur

7.1.3 Manuelle Eingabe mit IEyeDoc

IEyeDoc ist eine Webanwendung zur Eingabe von kompletten Datensätzen. Ein kompletter Datensatz besteht aus allen Patientendaten, den Visitendaten, Informationen zu Zusatzerkrankungen und deren Diagnosezeitpunkten, sowie den RNFLT Messungen. Mit dieser Anwendung kann man auch unvollständige Datensätze erweitern, sich Datensätze anzeigen lassen und Auswertungen zu den eingegebenen Daten ansehen. Wie in Abbildung 7.3 zu sehen ist, besteht die Anwendung aus mehreren Modulen. Die grafische Benutzeroberfläche wird von einer Silverlight 2.0 Webanwendung realisiert (Klassendiagramm in Abbildung 7.4). Diese kommuniziert mittels asynchroner Kommunikation mit einem WCF-Webservice (WCF steht für Windows Communication Foundation) dessen Klassendiagramm in Abbildung 7.5 zu sehen ist. Der WCF-Service nutzt das Binding „BasicHTTPBinding“ und kommuniziert mittels „LINQtoSQL“ mit der Datenbank. Er ist in C# geschrieben. Über den Service wird die komplette Datenbankanbindung realisiert. Der Webservice und die Silverlight-Anwendung sind auf einem Internet Information Service (IIS) auf einem Laptop installiert und die Datenbank läuft in einem Microsoft SQL Express 2008 Server, der ebenfalls lokal installiert ist. So findet keine Kommunikation nach außen statt.
Die Abbildung 7.6 verdeutlicht, worin die Aufgabe von IEyeDoc besteht und wie man mit der Anwendung arbeiten kann. Zur weiteren Veranschaulichung wird im nächsten Abschnitt noch ein kompletter Durchlauf durch die Oberfläche vorgeführt.

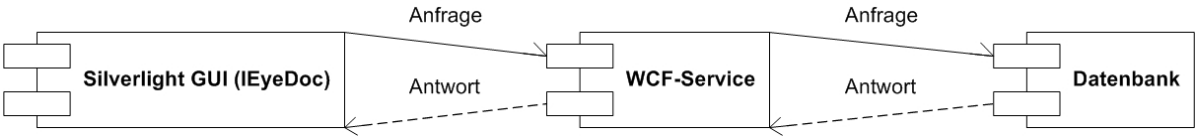


Abbildung 7.3: Grobe Architektur von IEyeDoc, dies soll die Komponenten und die Kommunikation zwischen den einzelnen Teilen besser veranschaulichen.

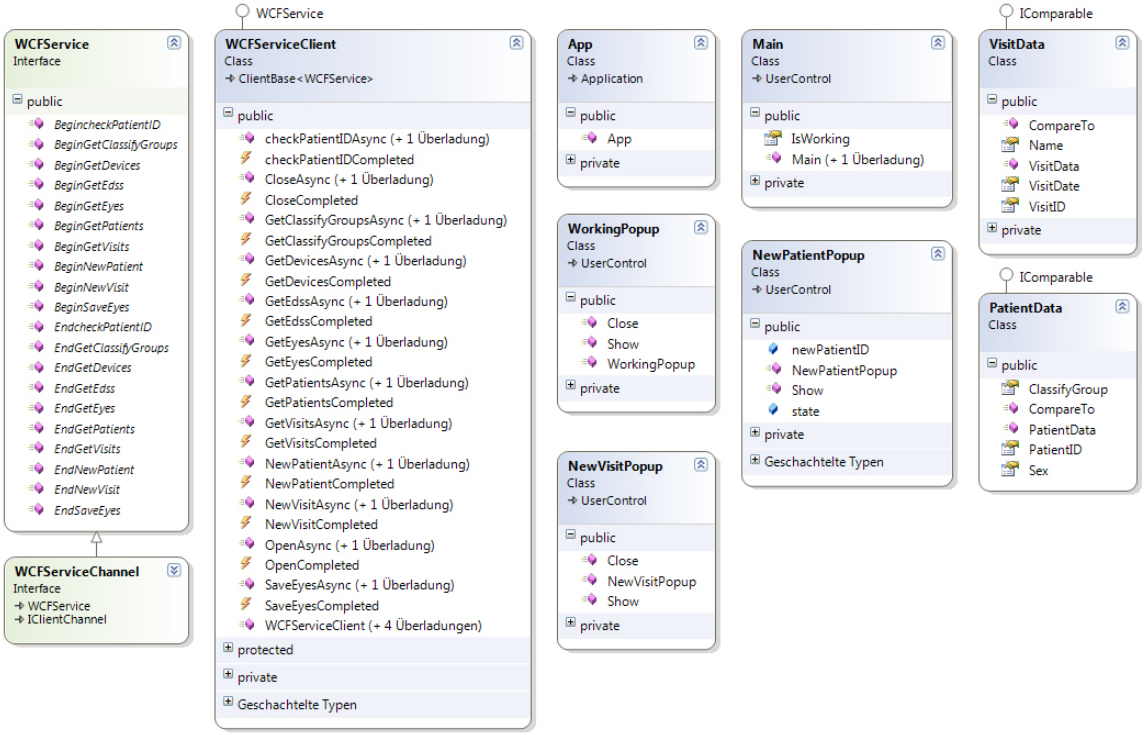


Abbildung 7.4: Klassendiagramm der Silverlight-Anwendung (IEyeDoc). Der WCFService stellt die Kommunikation bereit und die Silverlight-Oberfläche benutzt ihren SCFServiceClient, um diesen anzusprechen.

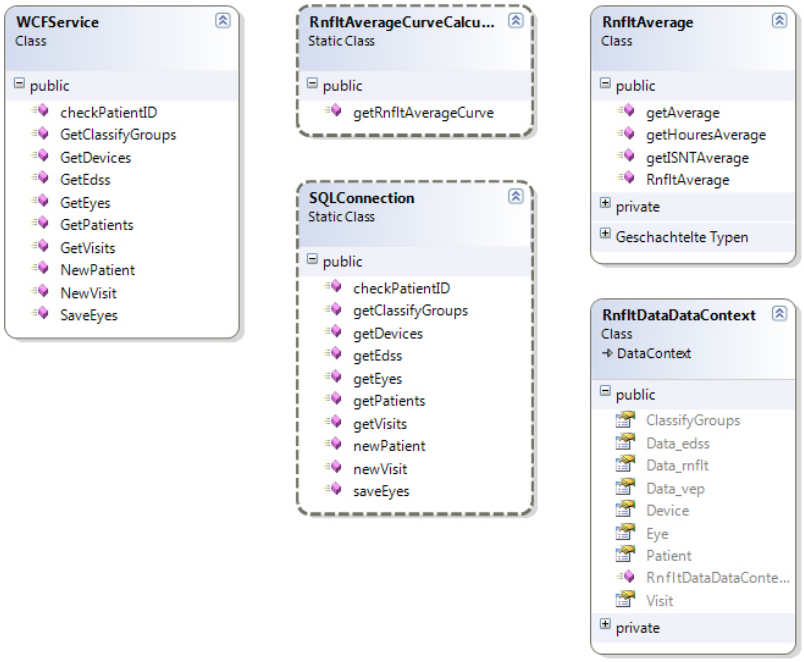


Abbildung 7.5: Klassendiagramm des WCF-Services von IEyeDoc

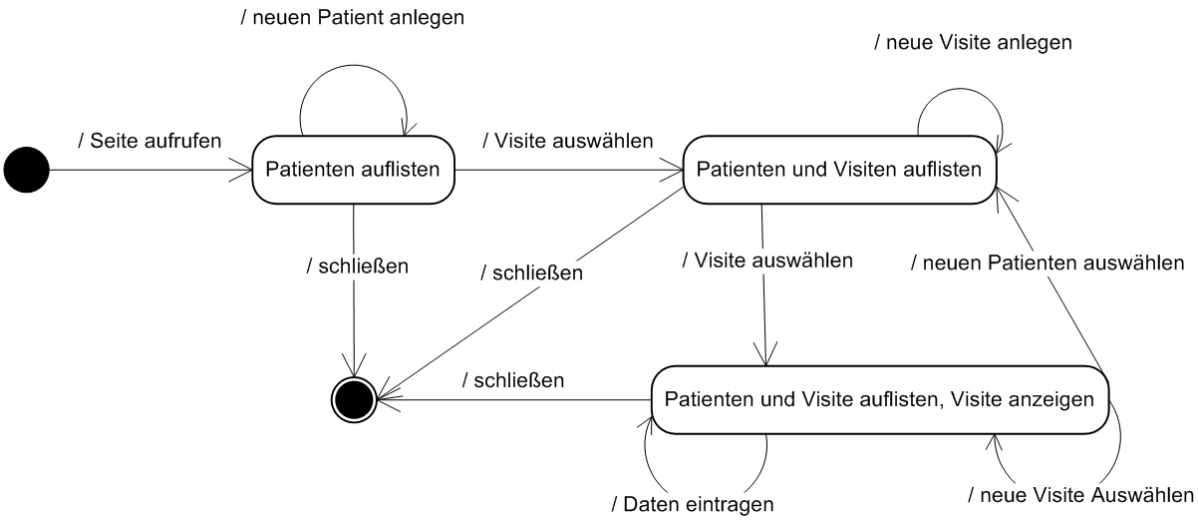


Abbildung 7.6: Zustandsdiagramm zu IEyeDoc, in dem die möglichen Abläufe dargestellt sind.

7.1.4 Überblick der Oberfläche von IEyeDoc

Mit dem Programm „IEyeDoc“ ist es möglich, wie in Abbildung 7.7 zu sehen ist, sich alle Informationen zu einer Visite auf einem Bildschirm anzeigen zu lassen. Ebenfalls können Patienten, Visiten, Augen und Messungen angelegt, angeschaut und verwaltet werden. Eine Präsentation der wichtigsten Funktionen soll das verdeutlichen.

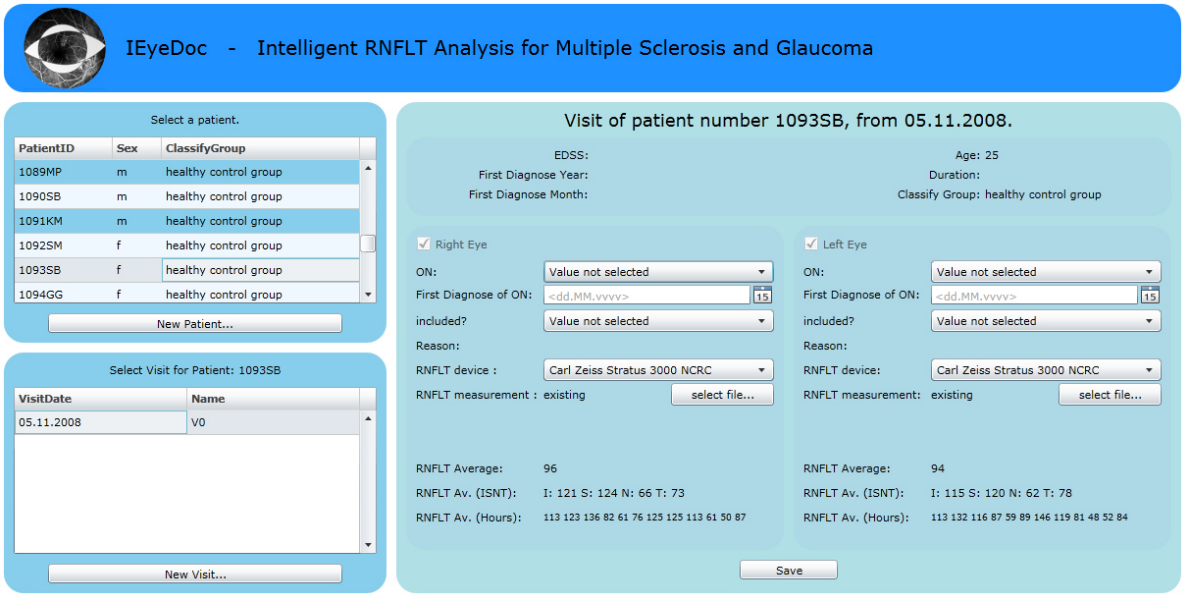


Abbildung 7.7: Screenshot von „IEyeDoc“ mit ausgewähltem Patient und Visite

Nach dem Start

Installieren braucht man „IEyeDoc“ nicht, das Programm ist über den Browser zu erreichen. Zur Zeit ist es auf einem Laptop in der Charité Berlin Mitte installiert und dort über einen lokalen Aufruf zu erreichen. Es ist auch im Browser als Startseite hinterlegt. Nach dem Start des Browsers gelangt man direkt zur Oberfläche von „IEyeDoc“, die sich wie in Abbildung 7.8 präsentiert.
Hier hat man nun die Möglichkeit, einen Patienten auszuwählen oder neu anzulegen.

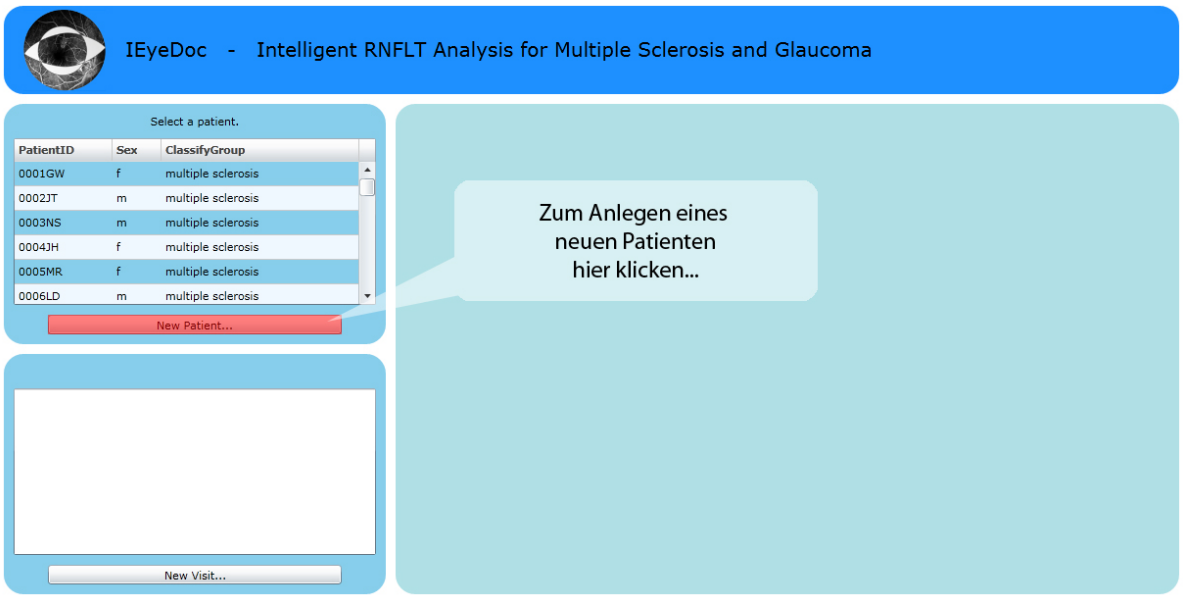


Abbildung 7.8: Screenshot von „IEyeDoc“ nach dem Starten

Anlegen eines neuen Patienten

Hat man auf „New Patient...“ geklickt, so erscheint eine Maske mit der Aufforderung, Daten zum Patienten einzugeben. Diese Maske ist in Abbildung 7.9 zu sehen. Nach dem Bestätigen des „create“-Button wird überprüft, ob alle Pflichtfelder ausgefüllt sind. Sollte ein Pflichtfeld vergessen worden sein, so erscheint ein Hinweis, ansonsten wird der Patient angelegt und man gelangt wieder zur vorherigen Seite. Mit dem Button „cancel“ kann man das Erstellen abbrechen und gelangt auch zur vorherigen Seite zurück.

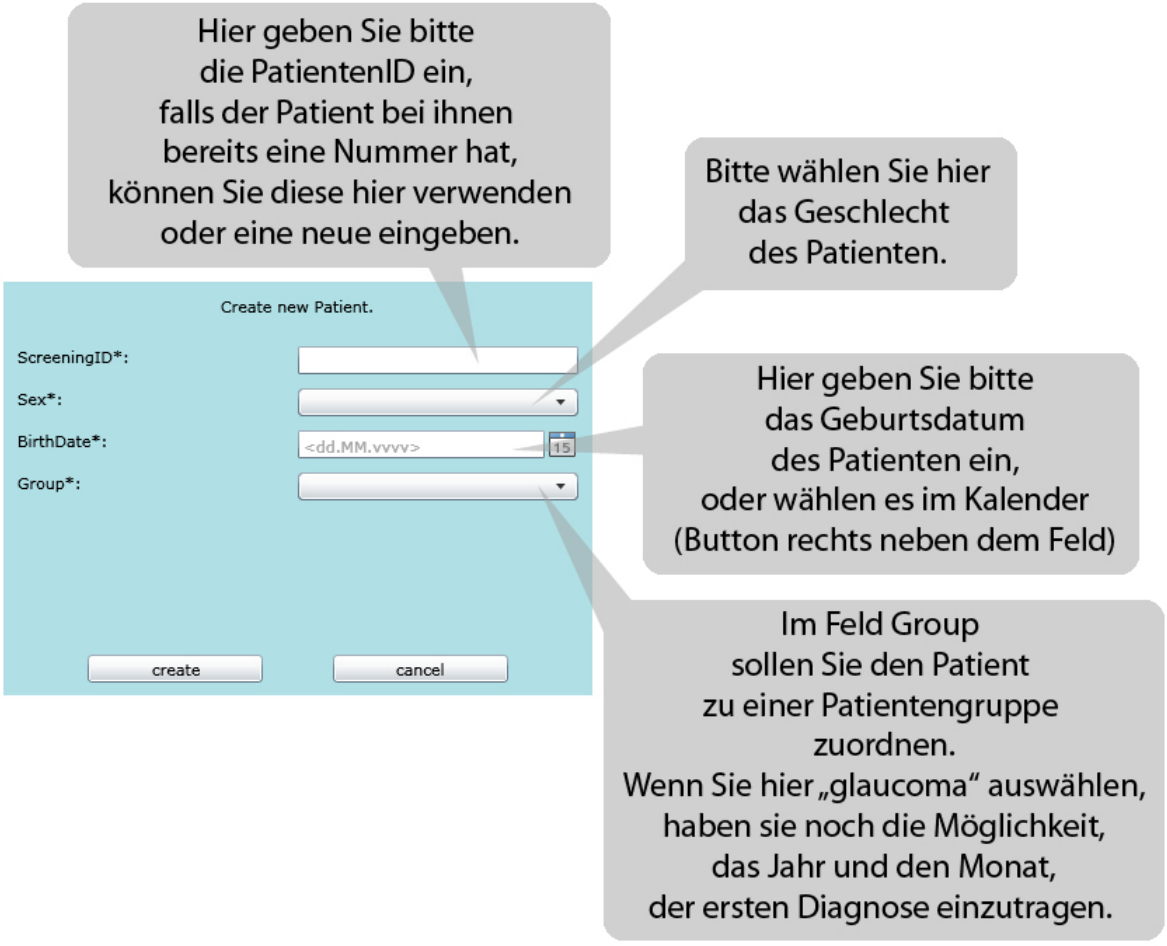


Abbildung 7.9: Screenshot von „IEyeDoc“, Maske zum Erstellen eines Patienten

Nach der Auswahl eines Patienten

Wenn ein Patient ausgewählt wurde, erscheinen in der Liste der Visiten alle zu dem Patienten gespeichert Visiten (siehe Abbildung 7.10). Von diesen kann eine ausgewählt oder eine neue Visite angelegt werden. Zu jedem Zeitpunkt ist es auch möglich, wieder einen anderen Patient auszuwählen.

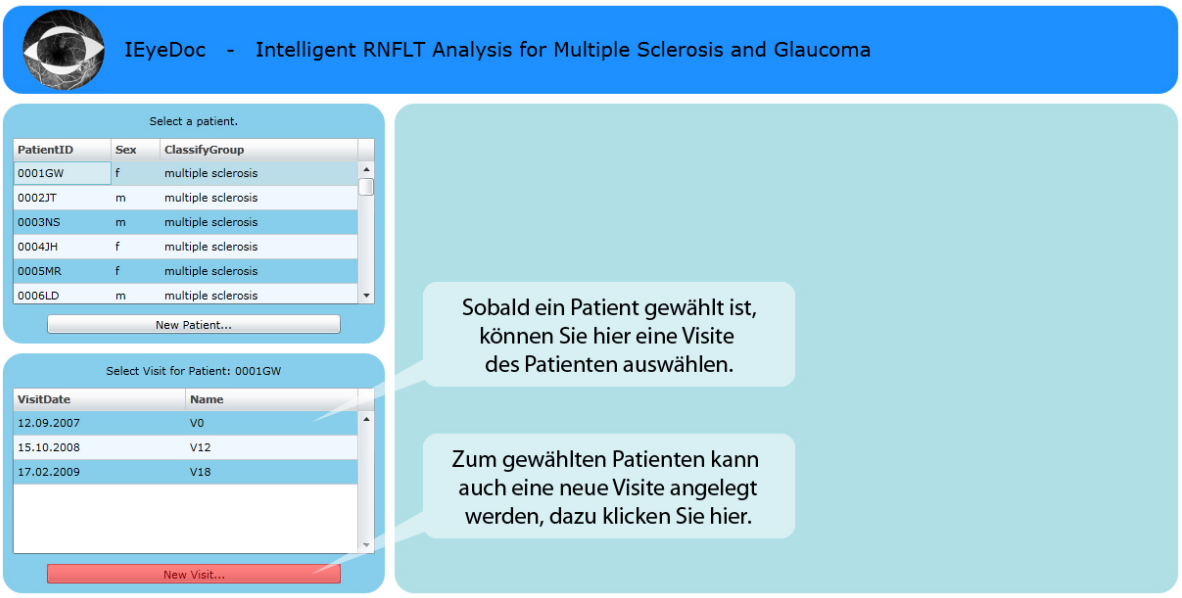


Abbildung 7.10: Screenshot von „IEyeDoc“, nach der Auswahl eines Patienten

Hinzufügen einer Visite

Nach dem Klicken auf den Button „New Visit...“ erscheint die Eingabemaske zum Erstellen einer neuen Visite, die in Abbildung 7.11 zu sehen ist. Hier werden alle Daten für eine Visite eingegeben. Sollte ein Pflichtfeld vergessen worden sein, so erscheint ein Hinweis. Wie auch bei der Eingabemaske für den Patienten, gelangt man mit „create“ und „cancel“ zur vorherigen Seite, wobei nur bei „create“ die Visite angelegt wird.

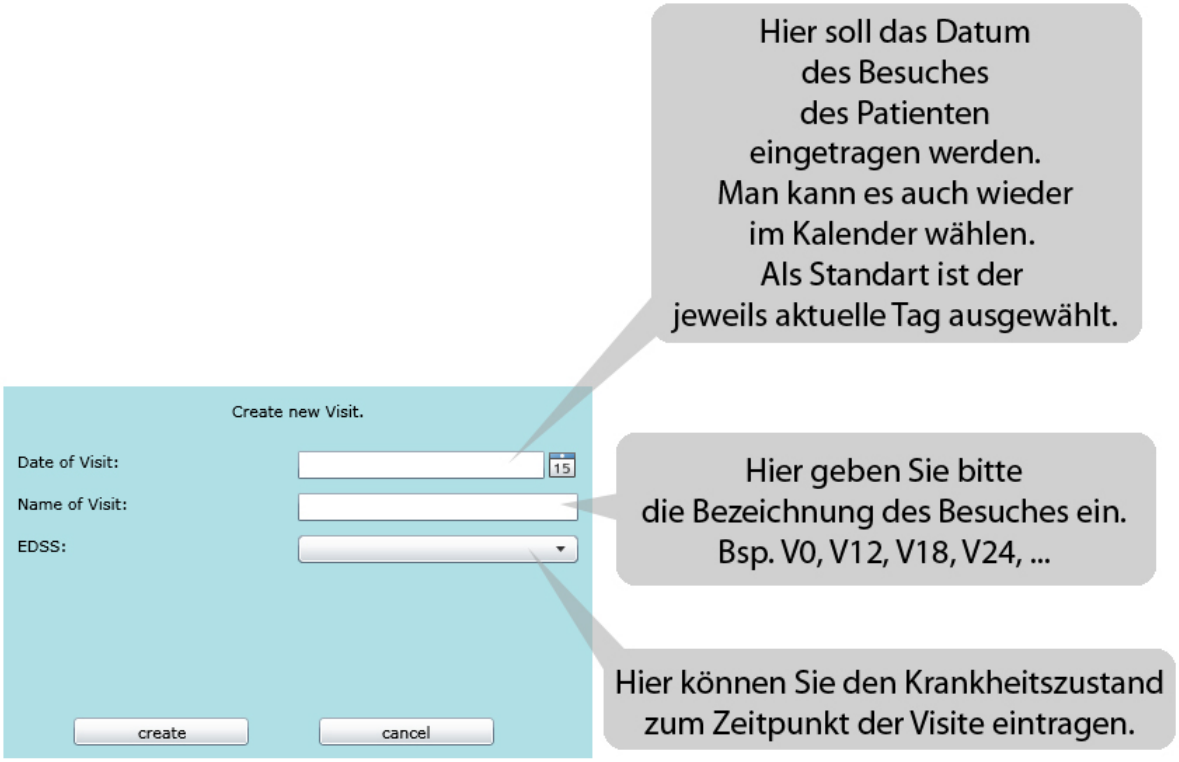


Abbildung 7.11: Screenshot von „IEyeDoc“, Maske zum Erstellen einer Visite

Visitenansicht

Wurde ein Patient und eine dazugehörige Visite ausgewählt, erscheint auf der rechten Seite die Visitenansicht, die wie in Abbildung 7.12 ausschaut. Hier stehen alle Daten, die zu den Augen und zu den Messungen eingegeben wurden oder noch eingegeben werden sollen. Einige kann man auch nachtragen oder verändern. Im oberen Bereich stehen die Patienten- und Visitenbezogenen Daten, im unteren Bereich die Daten, die sich auf ein Auge beziehen. Zu jedem Auge, welches über eine RNFLT-Messung verfügt, wird im unteren Teil der Anzeige auch eine kleine Auswertung der RNFLT-Messung angezeigt.

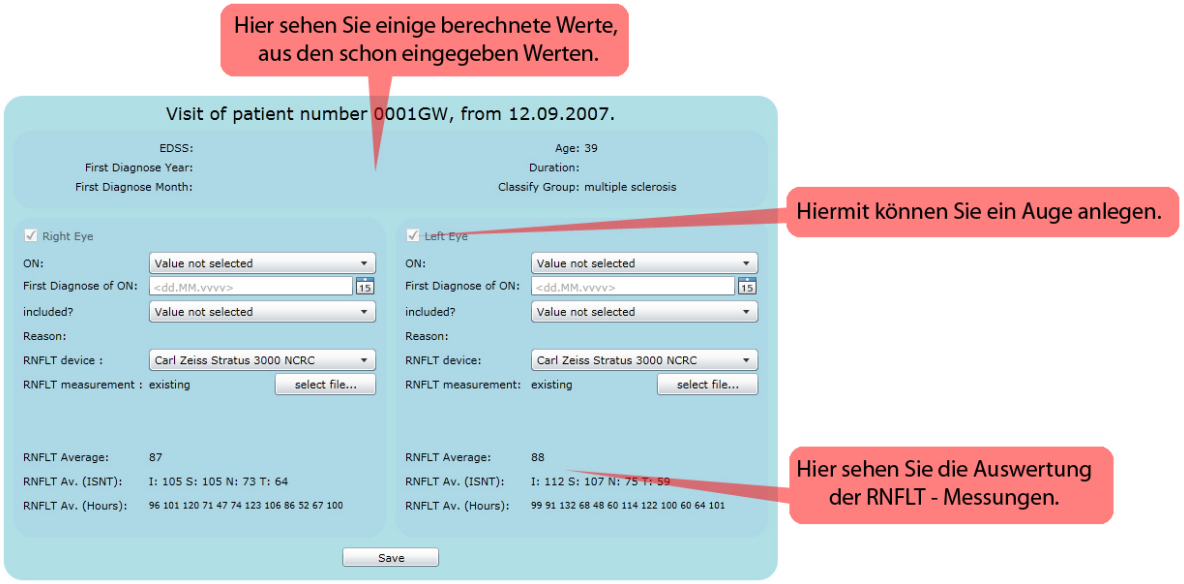


Abbildung 7.12: Zu sehen ist die Maske zum Anzeigen einer Visite von „IEyeDoc“. Oben sieht man einige Werte wie das Alter des Patienten und die Dauer der Erkrankung zum Zeitpunkt der Visite. Ebenfalls oben steht der Wert der Untersuchung EDSS, dieser spiegelt den allgemeinen Gesundheitszustand des Patienten wider. Im unteren Teil steht die erste Auswertung der RNFLT-Messung, hier wird der arithmetische Mittelwert der gesamten Messung, der Quadranten (I, S, N, T) und der Stunden (SN, S, ST, ..., NI, N, NS) angezeigt.

7.1.5 Automatischer Import mit RnfltImport

„RnfltImport“ ist ein Programm zum Einlesen der RNFLT-Messungen aus einem Verzeichnis. Dazu müssen die Messungen wie in [BB09] beschrieben exportiert und abgelegt werden. Von solchen abgelegten Messungen wird die Verzeichnisstruktur, die Textdatei und die RNF-Datei eingelesen. Dabei erfolgt eine Kontrolle der Messungen. Stimmt zum Beispiel die Verzeichnisstruktur nicht mit dem Dateinamen überein, so wird dort ein Fehler protokolliert. Dies geschieht auch wenn die Dateien in einem Verzeichnis nicht zueinander passen. Danach wird überprüft, ob die Messungen schon in der Datenbank existieren, falls nicht und es keine Fehler gab, werden sie hinzugefügt. Gab es Fehler, so kommen diese Datensätze nicht in die Datenbank und die Fehler sowie Erfolge werden nach dem Vorgang in einem Dialog aufgeführt. Der Ablauf des Programms ist noch einmal in Abbildung 7.14 veranschaulicht. Mit dem Programm können schnell sehr viele neue Messungen in die Datenbank übernommen werden.
Das Programm ist eine C# .NET Entwicklung, die mit der Version 3.5 in Microsoft Visual Studio 2008 umgesetzt wurde. Die Datenbankanbindung wurde mit LinqToSQL realisiert. Das Klassendiagramm des Programms ist in Abbildung 7.13 dargestellt.

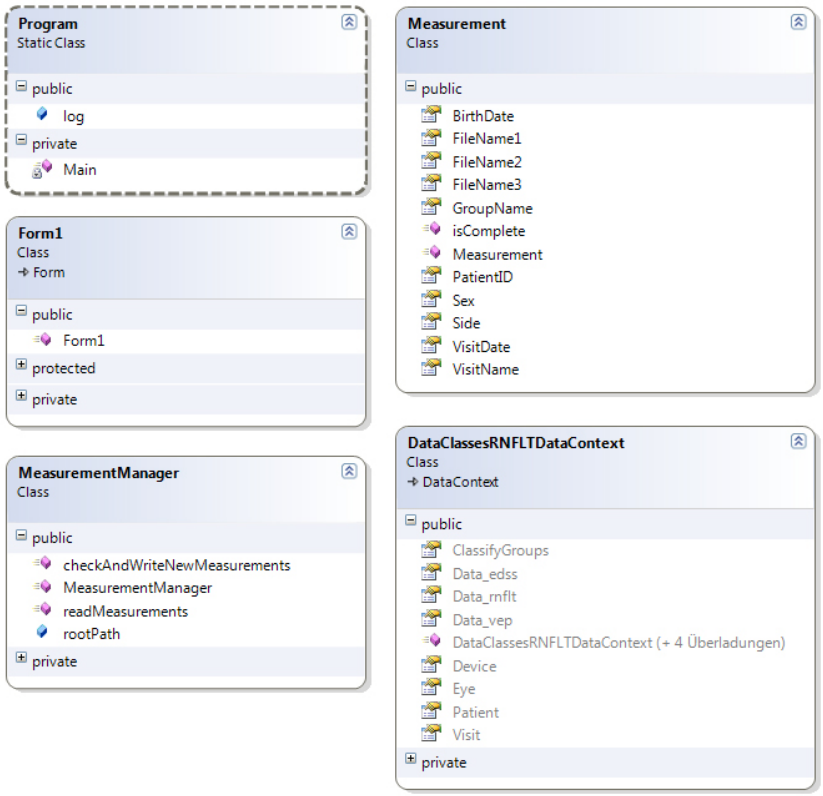


Abbildung 7.13: Klassendiagramm des Programms RnfltImport

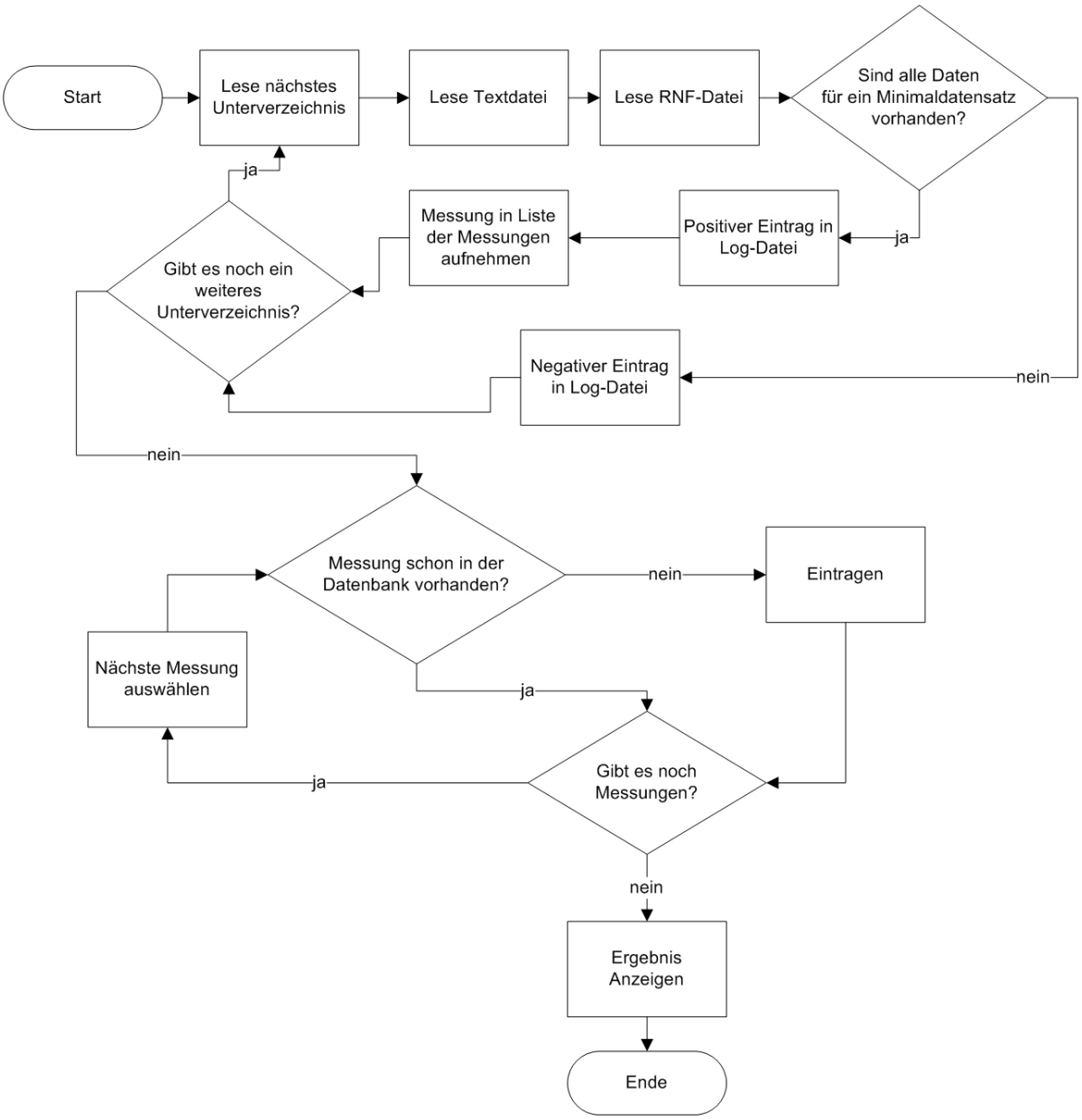


Abbildung 7.14: Ablaufplan des Programms „RnfltImport“

7.1.6 Überblick der Oberfläche von „RnfltImport“

Mit dem Programm „RnfltImport“ ist es möglich, RNFLT-Messungen schnell und einfach digital zu erfassen. Eine Präsentation der Funktionalität soll dies verdeutlichen.

Start von RnfltImport

Nach dem Start der Software erscheint das in Abbildung 7.15 zu sehende Fenster. Darin sieht man das Verzeichnis, aus dem die Messungen gelesen werden und einen Button, der den Vorgang startet.

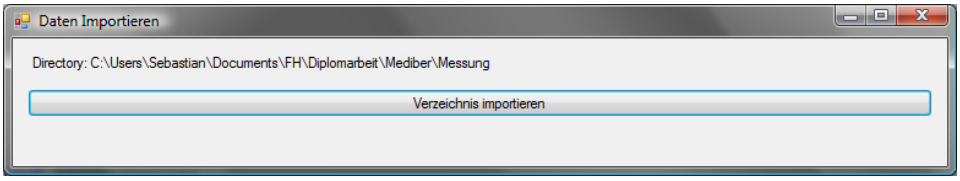


Abbildung 7.15: Screenshot von „RnfltImport“, nach dem Start

Fortschrittsanzeige

Da der Vorgang bis zu mehreren Minuten dauern kann, benötigt der Benutzer eine Rückmeldung. Diese Rückmeldung erhält er durch die Anzeige des grünen Fortschrittsbalken. Diesen sieht man in Abbildung 7.16.

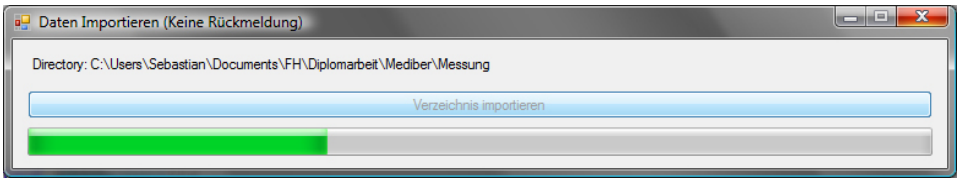


Abbildung 7.16: Screenshot von „RnfltImport“, während das Verzeichnis durchsucht wird

Anzeige des Ergebnisses

Der Nutzer sollte auch eine Rückmeldung über den Erfolg oder Misserfolg des Imports bekommen, dafür werden die Ergebnisse in einem Fenster nach Beendigung des Einlesevorgangs präsentiert, dieses Fenster sieht man in Abbildung 7.17.

Überprüfung der Log-Datei

Nachdem der Vorgang abgeschlossen ist und das Programm beendet wurde, kann man sich eine Log-Datei ansehen. Solch eine Datei sieht man in Abbildung 7.18. In der Log-Datei stehen mehr Informationen als in der Ausgabe zum Ende des Durchlaufs. Dies hilft bei der Analyse von Fehlern, die beim Import auftreten können. Meist sind dies Fehler, die beim Export der Messungen aus dem OCT-Gerät entstanden sind.

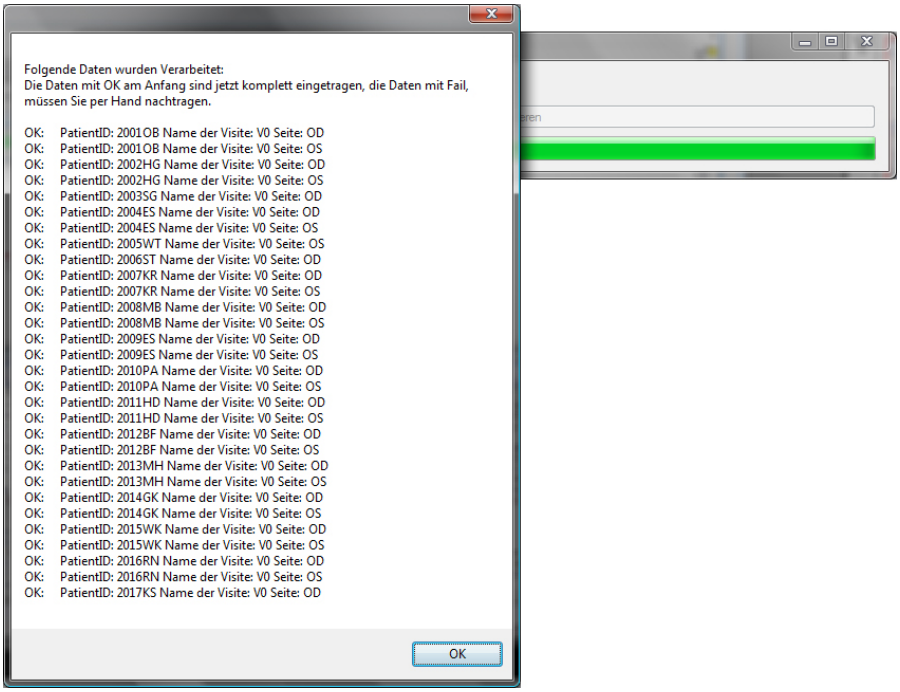


Abbildung 7.17: Screenshot von „RnfltImport“, mit Ausgabe nach dem Import

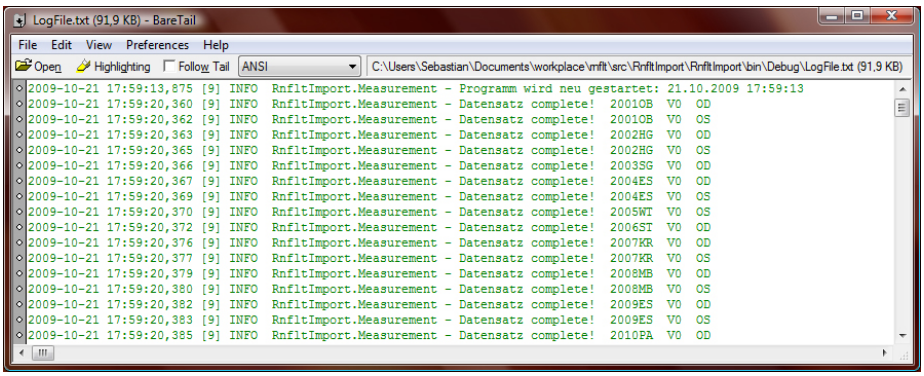


Abbildung 7.18: Screenshot der Logdatei von „RnfltImport“, nach einem Import

7.2 Datenbereinigung

Zur Datenbereinigung wurden mehrere Ansätze parallel verfolgt. Als erstes wurde darauf geachtet, dass bei der manuellen Eingabe die Eingabefelder hauptsächlich aus Auswahlfeldern bestanden. Nur in absolut nötigen Fällen wurden Freitexteingaben benutzt. Der zweite Ansatz betraf die automatische Erfassung der neuen Messungen. Dort wurde die Verzeichnisstruktur mit den Dateinamen und dem Inhalt der Textdateien verglichen und Redundanzen ausgenutzt, um die Korrektheit der Daten zu überprüfen. Sobald auch nur ein Wert nicht übereinstimmt, wird der Datensatz nicht eingetragen, sondern dem Benutzer mitgeteilt, dass er dies zu überprüfen hat. Der dritte Ansatz bezieht sich auf das manuelle Überprüfen der Datensätze von Fachkundigen. Dies waren in diesem Fall zwei Ärzte und ich selbst. Die Datensätze wurden von allen drei stichprobenartig durchsucht, ob doppelte Datensätze, Inkonsistenzen oder andere Fehler vorlagen. Dabei wurden auch drei Datensätze gefunden, bei denen es Unstimmigkeiten gab.

7.3 Datenerweiterung

Bei der Datenerweiterung sucht man neue Attribute, die sich aus den schon vorhandenen Daten ableiten lassen. Dies geschieht in diesem Fall durch ein Programm namens „RnfltAttributCalculator“.

7.3.1 RnfltAttributCalculator

Das Programm „RnfltAttributeCalculator“ liest die Datensätze der Datenbank aus und verarbeitet diese weiter. Es kann einzelne Datensätze anzeigen und berechnet neue Werte (Attribute) aus den vorhandenen Messreihen. Dieses Programm dient der Vorverarbeitung der Werte und der Erprobung neuer Attribute. Es ist möglich, sich die Originalwerte sowie alle berechneten Zwischenwerte anzeigen zu lassen. Dazu können auch Wertereihen in einem Diagramm angezeigt werden. Zur Anzeige der Diagramme wurde die externe Bibliothek „NPlot“ verwendet. Das Programm selber ist in C# .NET in der Version 3.5 in Microsoft Visual Studio 2008 als Windows Forms Applikation geschrieben. Um die Daten für das DataMining verwenden zu können, verfügt das Programm über die Möglichkeit, die erweiterten Datensätze in eine CSV-Datei zu schreiben. Dabei werden die Daten per Zufall in Lern- und Testdaten geteilt, wobei sichergestellt ist, dass in beiden Mengen der Anteil der Klassen gleich ist, die Mengen also stratifiziert sind. Das Klassendiagramm der Anwendung ist in Abbildung 7.19 zu sehen.

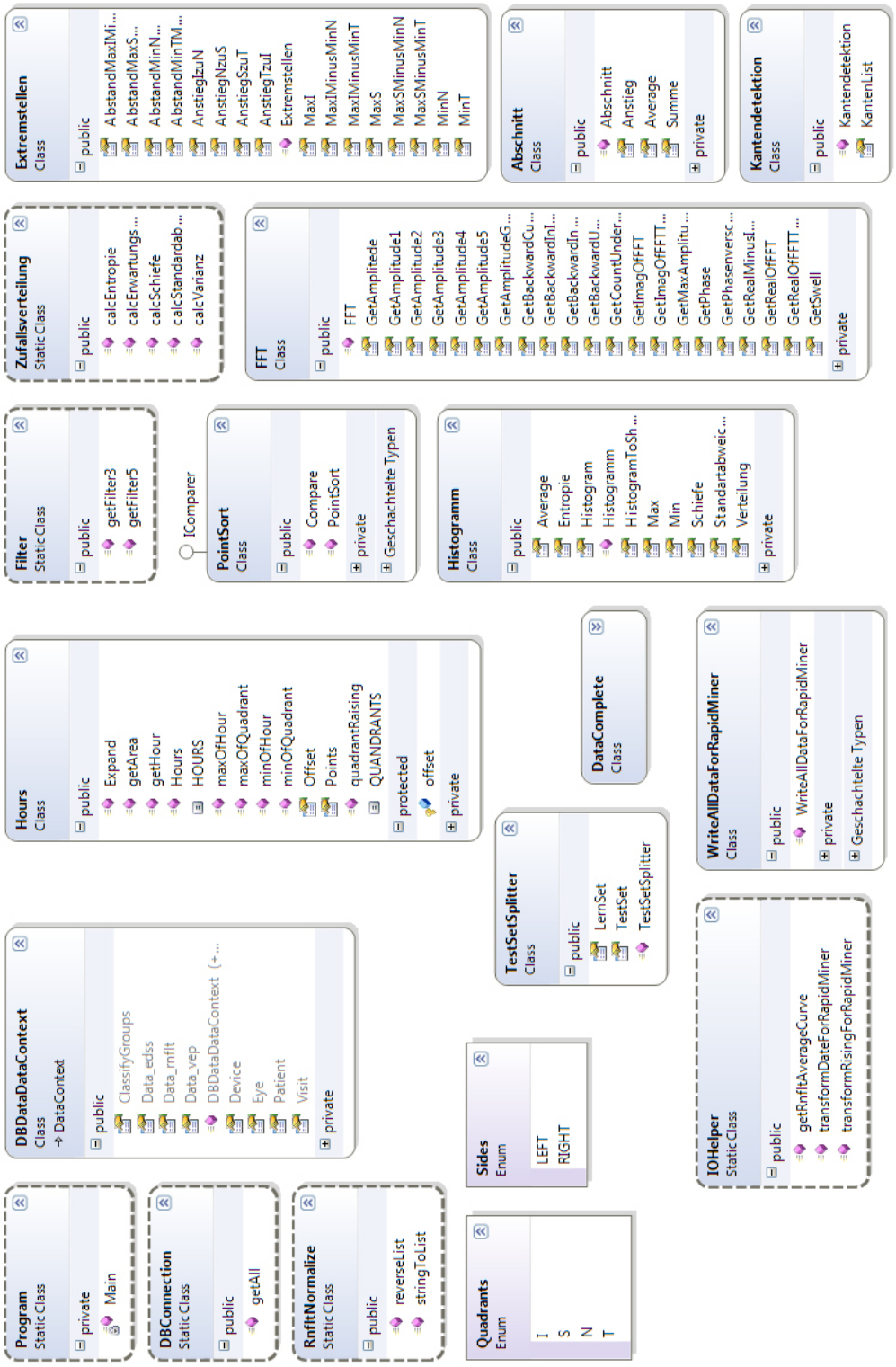


Abbildung 7.19: Klassendiagramm von „RnfltAttributeCalculator“. Rechts sieht man die Klassen, die Berechnungen auf den Daten durchführen, links unten die Klassen zum Schreiben der Daten und links oben die Klassen zur Kommunikation mit der Datenbank. Die Klasse „DataComplete“ sammelt alle Werte und ein Objekt dieser Klasse repräsentiert einen Datensatz.

7.3.2 Überblick der Oberfläche von RnfltAttributeCalculator

Wie „RnfltAttributeCalculator“ funktioniert und zu bedienen ist, zeigt der nächste Abschnitt.

Start von RnfltAttributeCalculator

Nach dem Start von „RnfltAttributeCalculator“ erscheint das in Abbildung 7.20 zu sehende Fenster. Hier ist noch nicht viel zu sehen, denn man muss erst die Daten aus der Datenbank laden.

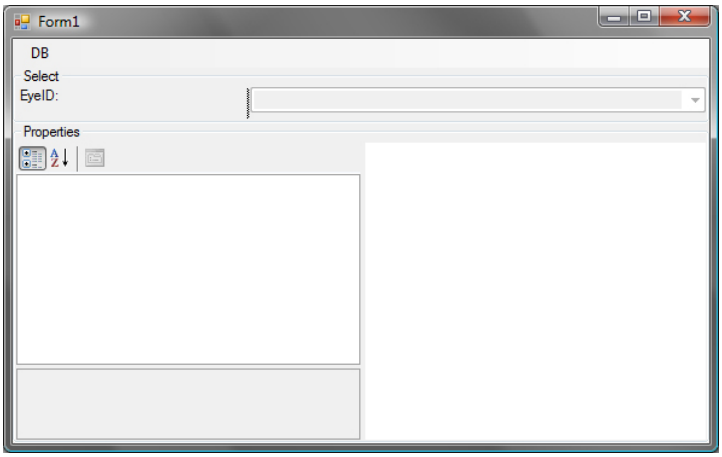


Abbildung 7.20: Screenshot von „RnfltAttributeCalculator“ nach dem Start

Laden der Daten

Um die Daten aus der Datenbank zu laden, muss man in der Menüleiste „DB“ auswählen und darunten den Punkt „GetAllComplete“ wählen. Dies ist in Abbildung 7.21 noch mal zu sehen. Nach dem Laden der Daten erscheinen die Identifikationszeichenketten im Auswahlfeld „EyeID“.

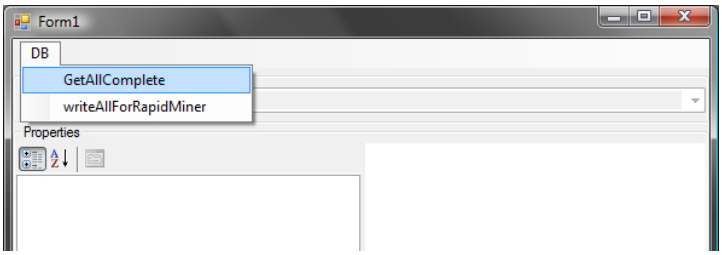


Abbildung 7.21: Screenshot von „RnfltAttributeCalculator“ mit aufgeklapptem Menü

Auswahl einer Messung

Von den IDs, welche wie in Abbildung 7.22 im Auswahlfeld aufgelistet sind, kann eine ausgewählt werden. Sobald dies geschehen ist, erscheinen alle zu der Messung gespeicherten Daten und die neu berechneten Attribute.

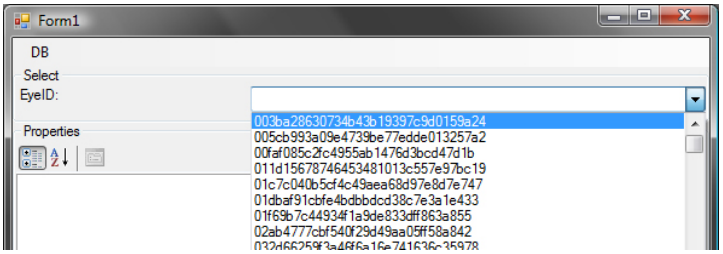


Abbildung 7.22: Screenshot von „RnfltAttributeCalculator“ mit gefülltem Auswahlfeld

Ansicht einer Messung

Im unteren Teil des Fensters erscheint nun die ausgewählte Messung. In Abbildung 7.24 sieht man, dass das Fenster zweigeteilt ist. Auf der linken Seite stehen die Attribute und ihre Werte. Auf der rechten Seite sieht man den durch „NPlot“ gezeichneten Graphen der Messreihe. Hier kann man auch noch viele andere Graphen einblenden. Durch das Auswählen einer anderen Augen-ID können Werte von Attributen so gut miteinander verglichen werden.

Export der Messungen für RapidMiner

Für die Auswertung ist es auch unbedingt notwendig, dass die neuen Attribute auch dem DataMining-Tool zur Verfügung stehen. Dafür beinhaltet das Programm eine Exportfunktion, mit der die Daten in ein für „RapidMiner“ lebares Format geschrieben werden können. Dazu muss man nur, wie in Abbildung 7.23 zu sehen ist, in der Menüleiste auf „DB“ klicken und darunter „writeAllForRapidMiner“ auswählen.

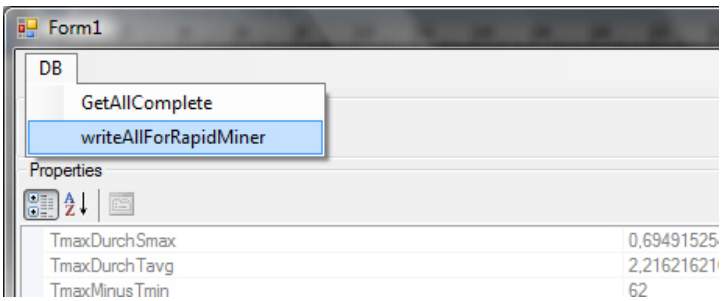


Abbildung 7.23: Screenshot von „RnfltAttributeCalculator“ mit dem Menü zum Exportieren der Messungen für RapidMiner

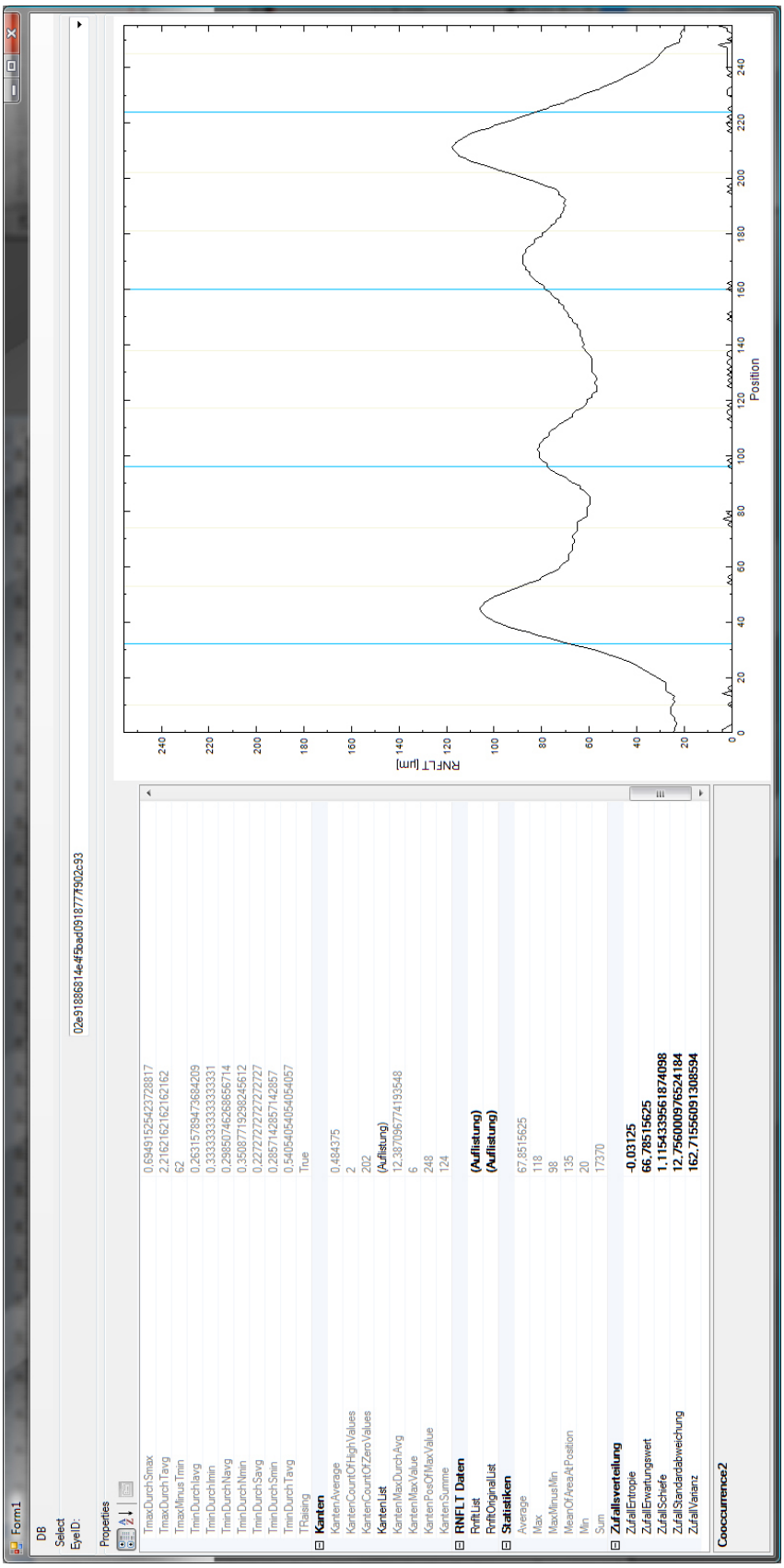


Abbildung 7.24: Screenshot von „RnfltAttributeCalculator“ mit der Ansicht eines Datensatzes zu einer Messung eines Auges

7.3.3 Berechnung neuer Attribute

An neue Attribute kann man über verschiedene Wege gelangen. Einer ist, einen Experten zu befragen und von ihm Hinweise zu bekommen, wie man an neue Attribute gelangen kann. Im besten Fall kann er sogar Regeln benennen, die angewendet werden können. Im Fall dieser Messungen waren die Hinweise darauf beschränkt, welche Bereiche der Messreihen markanter erschienen. Dieses Wissen wurde in den Quadranten und Stunden ausgewertet. Ein weiterer Weg ist das Betrachten von Statistiken, in diesem Fall wurden viele Messreihen nach geeigneten Attributen sortiert und miteinander verglichen. Dort fiel durch Anwenden von eigenem Wissen auf, dass Algorithmen der Bildverarbeitung, Mustererkennung und Fourier Transformation sinnvolle zusätzliche Attribute liefern sollten. Im Nachfolgenden werden nun die Ansätze und Berechnungen beschrieben.

Extremstellen und Mittelwerte

Über die gesamte Messreihe wird der Mittelwert, der Maximalwert und der Minimalwert gebildet. Diese Berechnungen werden auch auf die nachfolgend erläuterten Abschnitte angewendet.

Quadranten

Die Messreihe wird in vier Quadranten geteilt. Der erste liegt an der Nase, der zweite liegt am Ohr, der dritte an der Stirn und der letzte am Kinn. Aus der menschlichen Anatomie weiß man, dass die Nervenfasern in diesen Bereichen unterschiedlich dick übereinander liegen, weshalb diese Aufteilung sehr viel Sinn macht.
In den Bereichen werden auch sämtliche Berechnungen wie für die gesamte Messreihe durchgeführt.

Stunden

Die Aufteilung in vier Quadranten ist noch nicht alles. Einige Mediziner sind der Meinung, man kann auch unterschiedlich dick übereinander liegende Nervenfasern in noch kleineren Sektoren unterscheiden. Deshalb werden die vier Quadranten noch weiter in insgesamt 12 Stunden unterteilt und auch dort finden wieder die gleichen statistischen Berechnungen wie für die gesamte Messreihe statt.
Das Schwierige an dieser Aufteilung ist, zu bestimmen welcher Wert nun genau diese Bereiche trennt, da die Messreihe aus 256 Messwerten besteht und 256 nicht ganzzahlig durch zwölf teilbar ist. Dafür wurde die Messreihe so erweitert, dass jeder Wert nun dreimal hintereinander in einer neuen dreimal so langen Messreihe vorliegt. Dreimal, weil das kleinste gemeinsame Vielfache von zwölf und 256 die Zahl 768 ist und diese drei mal so viele Stellen wie 256 hat. Die erweiterte Messreihe hat nun eine Länge von 768 Werten und ist damit ganzzahlig durch zwölf teilbar.
In den Bereichen werden auch sämtliche Berechnungen wie für die gesamte Messreihe durchgeführt.

Spezielle Abschnitte

Bei der Betrachtung sehr vieler Messreihen ist auffällig, dass einige Abschnitte der Messreihen besonders markant erscheinen. Diese sind die Bereiche von Stelle 14 bis 31 und von 205 bis 220. Diese werden ab jetzt als „Abschnitt 1“ und „Abschnitt 2“ bezeichnet. Zu diesen wurde der Mittelwert und der Anstieg berechnet.

Histogramm

Das Histogramm ist die Häufigkeitsverteilung von Messwerten. Das heißt, wie oft jeder der 256 Messwerte in den Messreihen vorkommt.

Fourier Transformation

Die Fourier Transformation geschieht mit der externen Bibliothek „Math.Net“ [Rüe09], die Transformation gibt einen Real- und einen Imaginärteil einer reellen Zahl zurück, die in Amplituden-Phasen-Notation umgerechnet wird.

$$\text{Amplitude: } A_n = \sqrt{a_n^2 + b_n^2} \tag{7.1}$$

$$\text{Phase: } \varphi_n = 2arctan\frac{b_n}{A_n + a_n} \tag{7.2}$$

```
1 // FFT
2 RealFourierTransformation rft = new
     RealFourierTransformation();
3 rft.TransformForward(data, out outReal, out outImag);
4 // amplitude und phase
5 for (int i = 0; i < 128; i++)
6 {
7     amplitude[i] = Math.Sqrt(outReal[i] * outReal[i] +
        outImag[i] * outImag[i]);
8     phase[i] = Math.Atan2(outImag[i], outReal[i]);
9 }
```

Daraus wird der Amplitudenwert der Frequenz mit der maximalen Amplitude, ein Schwellwert (0,5 % der maximalen Amplitude), die Anzahl der Frequenzen mit einem Amplitudenwert unterhalb des Schwellwertes und der Wert der Phasenverschiebung der Frequenz mit der maximalen Amplitude berechnet. Vgl.[Kli01]

Kantenerkennung

Die Kantenerkennung versucht Sprünge oder Kanten in den Messreihen zu finden. Dazu wurde ein sehr spezielles Verfahren verwendet, das nach mehreren Versuchen das beste

Ergebnis hervorbrachte. Hierbei wird das Ergebnis der Fourier Transformation ausgenutzt. Bei der Fourier Transformation entstehen zwei Array, eines mit den Real- und eines mit den Imaginärteilen, daraus kann man ein Amplituden-Array und ein Frequenz-Array berechnen. Nun kann an jeder Stelle an der im Amplituden-Array der Wert größer als 2 % der maximalen Amplitude ist, den Wert im Real- und Imaginär-Array auf Null setzen. Danach führt man eine umgekehrte Fourier Transformation mit diesen Arrays aus, und erhält eine Messreihe in der nur noch die Störungen, Sprünge oder Kanten zu sehen sind. Die Störungen dieser Messreihe wurde dann noch mit einem Filter mit einer $(2, 0, -2)$ Maske verstärkt und mittels Betragsberechnung gleichgerichtet. Dies ist die Berechnung der Kantenerkennung. Im Quellcode sieht das wie folgt aus:

```
// Backward Curve
realInBackwardData = new double[256];
imagInBackwardData = new double[256];
outBackwardData = new double[256];
for (int i = 0; i < 256; i++)
{
   if (amplitude[i] > maxAmplitude * 0.02)
   {
      realInBackwardData[i] = 0;
      imagInBackwardData[i] = 0;
   }
   else
   {
      realInBackwardData[i] = outReal[i];
      imagInBackwardData[i] = outImag[i];
   }
}
TransformBackward(realInBackwardData, imagInBackwardData,
   out outBackwardData);
fftBackList = outBackwardData;

public Kantendetektion(List<double> fftBackList)
{
   List<int> list = new List<int>();
   List<int> filterList = new List<int>();
   List<int> helpList = new List<int>();
   for (int i = 0; i < 256; i++)
   {
      helpList.Add((int)fftBackList[i]);
   }
   filterList = Filter.getFilter3(helpList, 2, 0, -2, 0);
   for (int i = 0; i < filterList.Count; i++)
   {
```

```
33          list.Add(Math.Abs(filterList[i]));
34      }
35      KantenList = list;
36  }
```

In diesem Kanten-Array kann man nun die Stelle der stärksten Kante oder den Wert dieser Kante bestimmen und erhält somit gute Werte, die man vergleichen kann.

Cooccurrence-Matrix

Die Cooccurrence-Matrix ist ein zweidimensionales Feld $C[a, b]$, welches sich aus einem Ausgangs-Array $A[i]$ berechnet. Dabei wird das Array einmal komplett durchlaufen und für jedes $a = A[i]$ und $b = A[i + \Delta i]$ wird $C_{\Delta i}[a, b]$ um eins erhöht, wobei die Matrix vor Beginn mit Null initialisiert wurde.

- Im Quellcode sieht das wie folgt aus:

```
1  matrix = new int[list.Count, list.Count];
2  int i2;
3  // initialisieren der Matrix mit 0
4  for (int i = 0; i < list.Count; i++)
5  {
6      for (int j = 0; j < list.Count; j++)
7      {
8          matrix[i, j] = 0;
9      }
10 }
11 // Matrix erstellen
12 for (int i = 0; i < list.Count; i++)
13 {
14     i2 = i + abstand;
15     if (i2 > list.Count - 1)
16     {
17         i2 = i2 - list.Count;
18     }
19     matrix[list[i], list[i2]]++;
20 }
```

- Und als mathematische Formel:

$$C_{\Delta i}(a, b) = \sum_{i=0}^{255} \begin{cases} 1, & wenn\ A[i] = a\ und\ A[(i+\Delta i)\ mod\ 256] = b \\ 0, & sonst \end{cases} \tag{7.3}$$

Daraus werden dann verschiedene Attribute, wie die Stelle, mit dem maximalen Wert berechnet.

Zufallsverteilung

Das Histogramm einer Messreihe ist die Verteilung der Messwerte einer Messreihe. Fasst man diese nun als zufällige Verteilung auf, so kann man viele Berechnungen der Stochastik auf das Histogramm anwenden. Beispielsweise die Berechnung des Erwartungswertes, der Varianz, der Schiefe, der Standardabweichung oder der Entropie.

Der **Erwartungswert** ($E(x)$ oder μ) einer Zufallsvariablen (X) ist jener Wert, der sich bei mehrmaligem Wiederholen des zugrunde liegenden Experiments als Mittelwert der Ergebnisse ergibt. „Man erhält ihn als Mittel der Beobachtungswerte, wenn diese mit den entsprechenden Wahrscheinlichkeiten gewichtet werden."[Hüb03] In diesem Fall haben wir eine diskrete Menge von Zufallswerten und können deshalb die folgende Formel verwenden.

$$E(X) = \sum_i x_i p_i \tag{7.4}$$

Wobei $E(X)$ der Erwartungswert der diskreten Zufallsvariablen X mit den Werten $x_i, x_{i+1}, ..., x_n$ ist und p_i die Wahrscheinlichkeit des Auftretens von x_i. In dem Fall ist x_i der Wert der Messreihe an der Stelle i und $p_i = \frac{\text{Histogramm}[x_i]}{\text{Anzahl der Messwerte}}$. Daraus ergibt sich folgende Formel, wobei $H(x(i))$ das Histogramm der Messreihe an der Stelle i ist. vgl.[BHPT99]

$$E(X) = \sum_{i=0}^{255} x(i) \cdot \frac{H(x(i))}{256} \tag{7.5}$$

Die **Varianz** ist ein Maß, das beschreibt, wie stark eine Zufallsgröße um ihren Erwartungswert streut. Sie wird berechnet, indem man die Abstände der Messwerte vom Mittelwert quadriert, addiert und durch die Anzahl der Messwerte teilt. vgl.[BGG96]
Die Varianz für diskrete Zufallsgrößen berechnet sich nach der Formel:

$$V(X) = -(E(X))^2 + \sum_{i=0}^{255} (x_i)^2 \cdot p(x_i) \tag{7.6}$$

- $V(X) = Varianz$
- $E(X) = Erwartungswert$
- $p(x_i) = Wahrscheinlichkeit von x_i$

Die **Standardabweichung** ist ein Maß für die Streuung der Werte einer Zufallsvariablen um ihren Mittelwert. Sie ist für eine Zufallsvariable X definiert als die positive Quadratwurzel aus deren Varianz. vgl.[Bor89]
Die Standardabweichung berechnet sich nach der folgenden Formel, wobei $V(X)$ die Varianz der Zufallsvariablen X ist.

$$\sigma(X) = \sqrt{V(X)} \tag{7.7}$$

Die **Schiefe** beschreibt die „Neigungsstärke“ einer statistischen Verteilung X. Sie zeigt an, ob und wie stark die Verteilung nach rechts (positive Schiefe) oder nach links (negative Schiefe) geneigt ist. Ist der Wert der Schiefe kleiner Null, so ist die Verteilung *linksschief*, ist sie größer Null, so ist die Verteilung *rechtsschief*. Und es steckt noch mehr in dem Attribut, je weiter der Wert an Null liegt, desto symmetrischer ist die Messwertreihe. vgl.[BHPT99]
Sie berechnet sich nach Formel 7.8 mit $E(X)$ = Erwartungswert, X = Zufallsvariable, $\sigma(X)$ = Varianz der Zufallsvariablen.

$$v(X) = \frac{E((X - E(X))^3)}{\sigma^3(X)} \tag{7.8}$$

Für den Anwendungsfall in dieser Arbeit kann man die Formel wie folgt schreiben.

$$v(X) = \frac{\sum_{i=0}^{255} ((x_i - E(X))^3 \cdot p(x_i))}{\sigma^3(X)} \tag{7.9}$$

Die **Entropie** ist ein Maß für den mittleren Informationsgehalt oder auch Informationsdichte eines Zeichensystems. Der Begriff Entropie stammt ursprünglich aus der Informationstheorie und wird hier etwas abgewandelt verwendet. Er kann hier aber verwendet werden, da man die Messreihe auch als übertragenes Signal interpretieren kann. vgl.[Joh92]
Die Entropie H einer diskreten Zufallsvariable mit dem Alphabet $Z = \{z_1, z_2, ..., z_n\}$ berechnet sich wie folgt:

$$H = \sum_{z \in Z} p_z \cdot I(z) = - \sum_{z \in Z, p_z \neq 0} p_z \cdot log_2(p_z) \tag{7.10}$$

wobei $I(z) = -log_2(p_z)$ der Informationsgehalt und p_z die Wahrscheinlichkeit von z ist. Die Bedingung $p_z \neq 0$ ergibt sich durch die Regel von l'Hospital, denn

$$\lim_{p_z \to 0} p_z \cdot log_2(p_z) = 0. \tag{7.11}$$

Für den Anwendungsfall in dieser Arbeit ist das Alphabet $Z = \{0, 1, ..., 255\}$, die Wahrscheinlichkeit

$$p_z = \frac{\text{Histogramm[z]}}{\text{Anzahl der Messwerte}} \tag{7.12}$$

und die Berechnung der Entropie dann:

$$H = - \sum_{z=0, \text{Histogramm}_{[z] \neq 0}}^{255} \frac{\text{Histogramm[z]}}{\text{Anzahl Messwerte}} \cdot log_2 \left(\frac{\text{Histogramm[z]}}{\text{Anzahl der Messwerte}} \right). \tag{7.13}$$

7.4 Datenauswertung

Zur Datenauswertung wurde das Programm „RapidMiner 4.5 Community Edition“ der Firma „Rapid-I GmbH“ mit den Datensätzen aus „RnfltAttributeCalculator“ benutzt.

7.4.1 Erstellung eines Setup

Das grobe Setup ist jenes aus der Vorüberlegung geblieben. Die Aufteilung der Lerndatensätze erfolgt durch eine zehnfache X-Validierung. In Abbildung 7.25 sieht man ein Schema des Setups. Hier wird nur noch die Auswahl der zu untersuchenden Attribute und der Lernalgorithmus verändert. Im Einzelnen geschieht nun Folgendes.
In „LernExampleSource“ werden die Lerndaten geladen. An dieser Stelle werden verschiedene Attributauswahlen getestet. „XValid“ ist die X-Validierung, welche die Daten noch einmal in zehn Teile teilt. In zehn Versuchen wird jeweils einer als Testmenge und die anderen als Lernmenge benutzt. Der „LibSVMLerner“ steht hier als Platzhalter für die Lernalgorithmen, in denen an Hand der Lernmenge ein Modell erlernt wird. Dieses wird vom „Applier“ auf die Testmenge angewendet, somit entsteht eine Vorhersage der Klasse. Diese wird von dem mit „BinominalClassificationPerformence“ benannten Algorithmus bewertet. Nach der X-Validierung liegen zehn solche Bewertungen vor, von denen der Mittelwert gebildet wird, welcher als erstes Ergebnis dient. Nun wird der gleiche Lernalgorithmus erneut auf alle Lerndaten angewendet. Das hieraus entstandene Modell wird anschließend auf die neu geladenen Testdaten angewendet. Hierbei entsteht wieder eine Vorhersage. Die Bewertung dieser Vorhersage bildet das zweite Ergebnis.

7.4.2 Auswahl der Eingangsattribute

Im Laufe der Auswertung ist aufgefallen, dass es sinnvoll erscheint, die Auswertung nicht immer mit allen Attributen durchzuführen. Eine Selektion der Attribute macht gerade zum Vergleichen der verschiedenen Ansätze Sinn. Dadurch wird auch die Berechnungszeit kürzer und die Ergebnisse teilweise besser.
Für eine Auswahl der Attribute konnte man verschiedene Ansätze nutzen. Entstanden sind vier Auswahlen, die in Tabelle 7.3 aufgeführt sind.
Die Mittelwerte wurden schon öfter, auch von anderen Studien ausgewertet, weshalb diese Auswahl als Referenz dienen kann.
Die Auswahl nach dem Chi-Quadrat-Test reduziert die Menge der Attribute auf diejenigen, die auch allein schon eine gute Trennung zwischen den Klassen schaffen könnten. Einen Auszug aus den Ergebnissen des Tests ist in Abbildung 7.26 dargestellt. Je höher ein Wert ist, desto besser ist dieses Attribut zur Unterscheidung der Klassen geeignet. Die Werte reichen von 0,0 bis 1,0. Für die Berechnung müssen alle Attribute diskret sein, deshalb wird jedes kontinuierliche Attribut auf maximal zehn diskrete Werte gekürzt.
Die letzte Auswahl besteht aus den Mittelwerten und einigen zusätzlichen guten Attributen, hier sollte sich eine Verbesserung zu der Auswahl zeigen, welche nur die Mittelwerte beinhaltet.

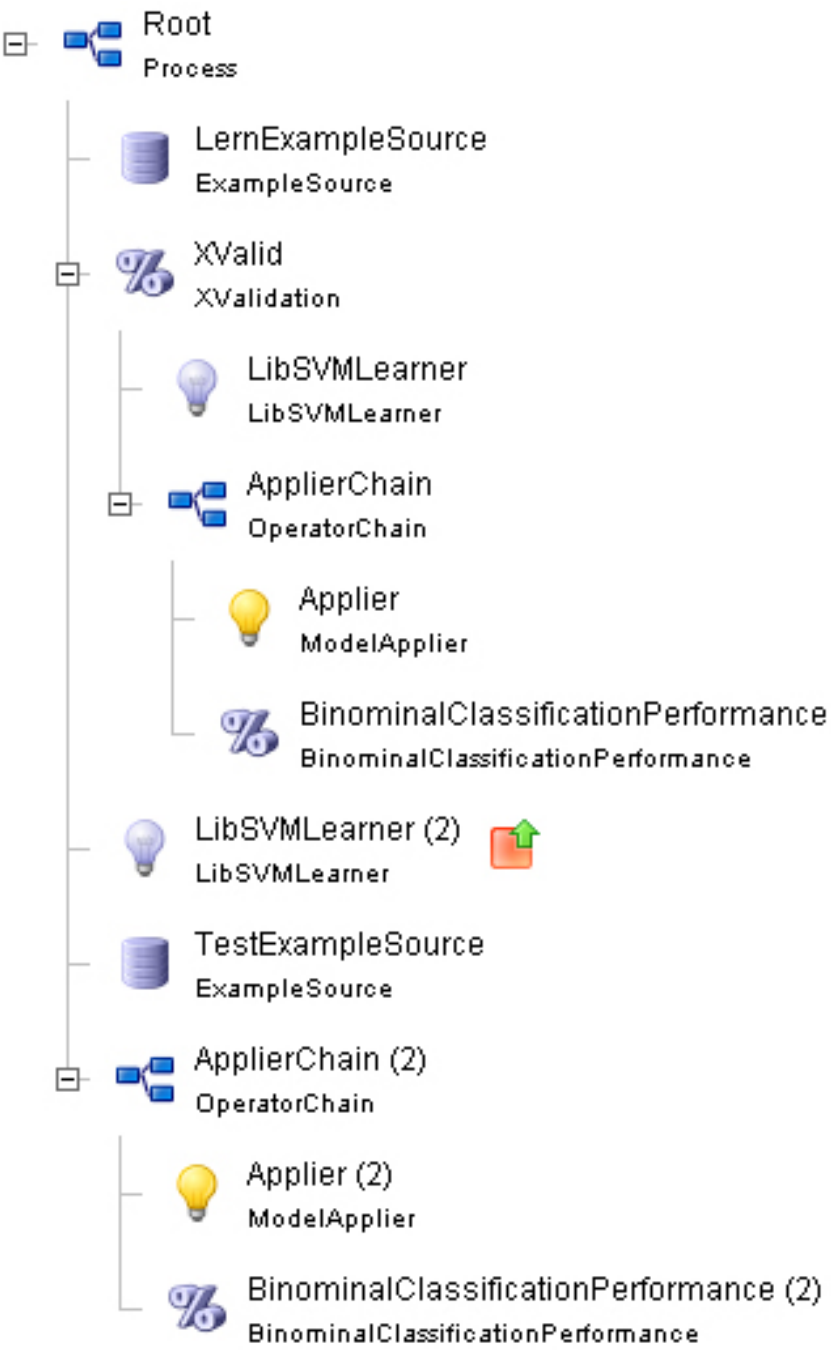


Abbildung 7.25: Setup eines DataMining Versuchs

Auswahlnummer	**Attribute der Auswahl**
1	Alle zur Verfügung stehenden Attribute
2	Nur die arithmetischen Mittelwerte der gesamten Messreihe sowie von allen Quadranten und Stunden
3	Alle Attribute, die bei einer Gewichtung mit dem Chi-Quadrat-Test einen Wert zwischen 0,5 und 1,0 erreicht haben. Das Ergebnis des Chi-Quadrat-Test sieht man in Abbildung 7.26.
4	Alle Mittelwerte sowie eine Auswahl von frei gewählten einigen Attributen aus jedem Ansatz. Diese sind in der folgender Liste aufgelistet. • Sex • AgeAtVisit • GesamtMin • GesamtMax • HistogrammMax • FFTMaxAmplitude • Bereich1Avg • Bereich2Avg • ZufallErwartungswert • MaxI.X • MaxS.X • MinN.X • MinT.X • KantenAverage

Tabelle 7.3: Auflistung der Eingangsattributauswahlen für das DataMining Setup

attribute	weight
GesamtAverage	1
GesamtMin	0.841
MinValueOfHourTI	0.835
AverageHourT	0.801
AverageQuadrantT	0.730
GesamtMax	0.727
AverageHourTI	0.725
MinValueOfHourTS	0.720
AgeAtVisit	0.693
MaximumQuadrantI	0.688
MinimumQuadrantT	0.686
MinValueOfHourT	0.683
MaximumQuadrantS	0.680
Bereich1Avg	0.655
MaxValueOfHourTI	0.643
AverageQuadrantS	0.643
MaxValueOfHourIT	0.639
MaxValueOfHourT	0.639
MinValueOfHourST	0.618
AverageHourST	0.610
MaxValueOfHourST	0.608
AverageQuadrantI	0.603
MinimumQuadrantI	0.598
Bereich2Avg	0.594
AverageHourTS	0.588
MinValueOfHourIT	0.570
AverageHourIT	0.562
MaxValueOfHourS	0.545
MaximumQuadrantT	0.531
MinimumQuadrantS	0.522

Abbildung 7.26: Alle Attribute mit Ergebnis des Chi-Quadrat-Tests, welche besser als mit 0,5 abgeschnitten haben.

7.4.3 Lernalgorithmen

Es werden vier unterschiedliche Lernalgorithmen verglichen. Der erste Lernalgorithmus ist ein „DefaultLerner“, er ist eigentlich kein Lernalgorithmus, denn er sagt nur eine Klasse voraus. Er dient nur dazu einen Ausgangswert für die Bewertung zu haben. Ein erster richtiger Lernalgorithmus ist der „Naive Bayes“-Algorithmus, der mit Standardeinstellungen verwendet wird. Dieser basiert auf der Berechnung von Wahrscheinlichkeiten jedes beteiligten Attributes. Ein zweiter verwendeter Lernalgorithmus ist eine „Support Vector Machine“ namens „LibSVMLerner“ mit linearem Kernel. Und der dritte untersuchte Algorithmus ist auch diese „Support Vector Machine“, allerdings mit polynomen Kernel fünften Grades. Die Einstellungen an den Algorithmen wurden festgelegt, nachdem viele Test durchgeführt wurden und sich durch Annäherung diese Einstellungen als optimal erwiesen haben.

7.4.4 Bewertungsalgorithmen

Für die Bewertung der Modelle standen einige Bewertungsparameter zur Verfügung. Durchgesetzt haben sich die Fläche unter der ROC-Kurve (engl.: area under curve, AUC) und die Sensitivität, bei einer gegebenen Spezifität von 95 % und 98 %. Diese Werte sind bei klinischen Studien sehr gefragt und sollten deshalb immer mit angegeben werden. Aus diesem Grund sollen sie auch die entscheidenden Werte bei dieser Untersuchung sein.
Den Wert von AUC berechnet RapidMiner, dieser braucht also nur abgelesen zu werden. Die beiden Sensitivitäten kann man mit RapidMiner nicht direkt berechnen lassen. Dazu wird eine ROC-Kurve berechnet, aus der die Werte abgelesen werden müssen. Eine solche ROC-Kurve sieht man in Abbildung 7.27.

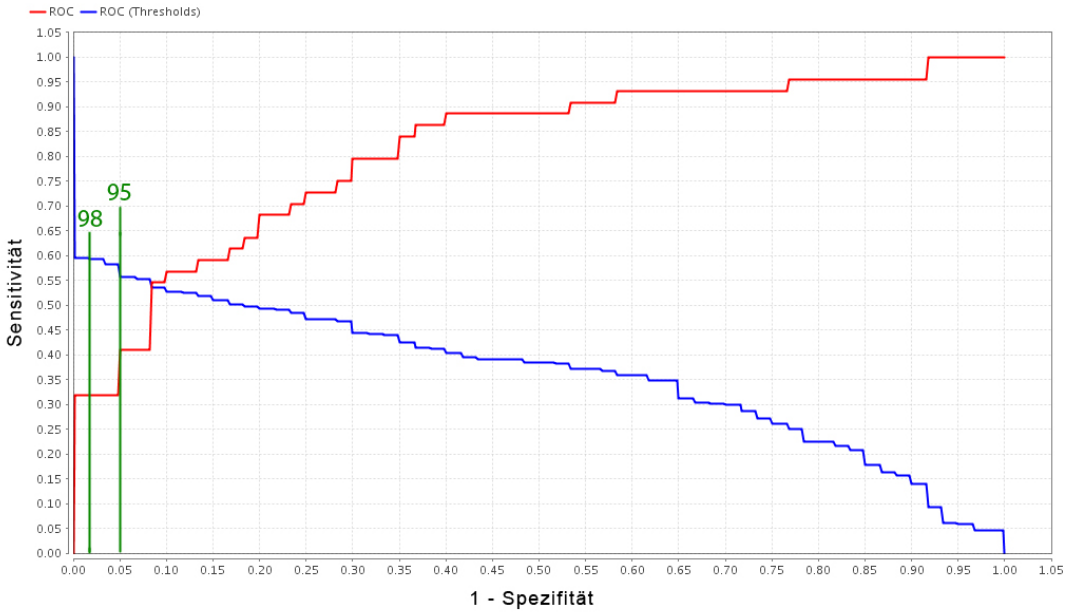
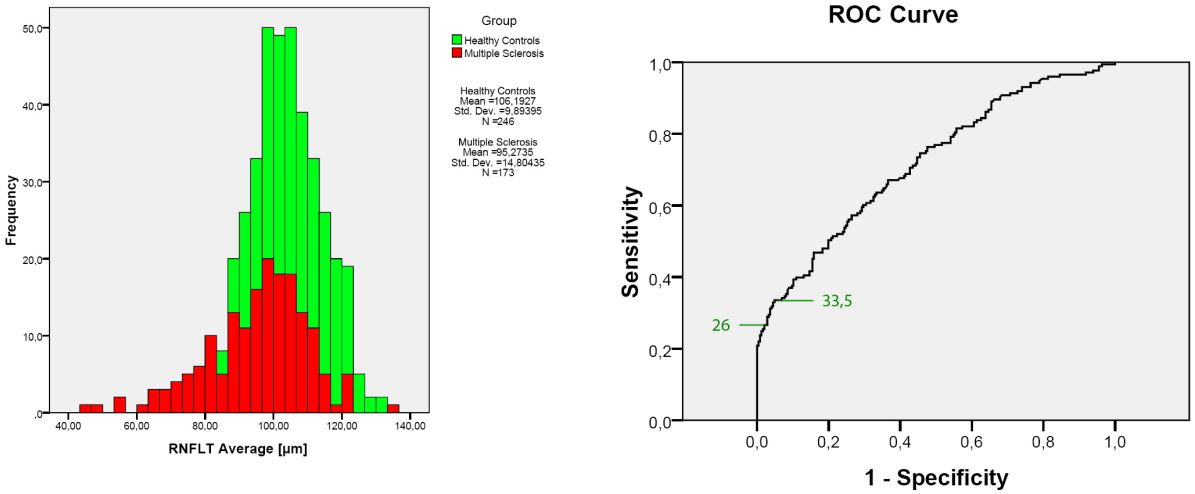


Abbildung 7.27: Zu sehen ist eine typische ROC-Kurve, die rote Kurve ist die ROC-Kurve, die blaue Kurve ist die Kurve mit den Schwellwerten. Grün markiert sind die Stellen, an denen eine Spezifität von 95 % und 98 % vorliegt.

7.4.5 Referenzauswertung

Bevor man sich jetzt die Ergebnisse dieses Verfahrens betrachtet, sollte man Referenzwerte kennen, an denen man dieses neue Verfahren messen kann. Wie schon weiter vorn erwähnt, ist zur Zeit die Auswertung des arithmetischen Mittelwerts der Messreihe, die bevorzugte Art und Weise solche Messungen miteinander zu vergleichen. Solch eine Auswertung sieht man in Abbildung 7.29, sie stammt aus einer bislang noch nicht veröffentlichten Studie von Markus Bock [BBD+09], der diese Daten zur Verfügung gestellt hat. Die Auswertung beruht auf Daten einer Alters- und Geschlechtsübereinstimmenden Datenbank von gesunden und erkrankten Probanden. Solch gute Übereinstimmung liegt bei den in dieser Arbeit verwendeten Probanden nicht vor, weshalb der Vergleich nicht hundertprozentig korrekt sein wird.
Um die Vergleichbarkeit zu realisieren, wurde die Auswertung des Mittelwertes auch auf die für diese Arbeit zu Grunde liegenden Daten angewandt. Dazu gibt es eine Grafik zur Verteilung der Klassen und die daraus abgeleitete ROC-Kurve, sowie eine Tabelle mit dem Wert der Fläche unter der ROC-Kurve (AUC). Die beiden Grafiken und die Tabelle können in Abbildung 7.28 betrachtet werden. Die Sensitivität bei einer Spezifität von 95 % beträgt in diesem Fall 33,5 % und bei einer Spezifität von 98 %, beträgt die Sensitivität 26 %. In der Tabelle ist der Wert „Area“ aufgeführt, dies ist die Fläche unter der ROC-Kurve.
Auch diese Werte sind nicht hundertprozentig zum Vergleich der Ergebnisse geeignet,

denn die Referenzauswertung bezieht sich auf alle Datensätze. Die in dieser Arbeit vorgestellte Methode aber auf mehrere Datensätze, wobei die Lern- und Testdatensätze immer strickt von einander getrennt waren.
Die zwei Sensitivitäten und der Wert von AUC sollen trotz allem als Referenzwerte für das neue Verfahren dienen, denn bessere Vergleichswerte gibt es leider nicht.



Test Result Variable(s):RNFLT Average [µm]

Area	Std. Errora	Asymptotic Sig.b	Asymptotic 95% Confidence Interval	
			Lower Bound	Upper Bound
,720	,025	,000	,670	,769

The test result variable(s): RNFLT Average [µm] has at least one tie between the positive actual state group and the negative actual state group. Statistics may be biased.

a. Under the nonparametric assumption

b. Null hypothesis: true area = 0.5

Abbildung 7.28: Ergebnisse der Auswertung des Average-Wertes der Messreihe. Links die Verteilung der Werte in den Klassen, rechts die resultierende ROC-Kurve und unten eine Tabelle mit dem AUC-Wert und der Signifikanz.

Table 1: RNFLT measurement differences from Multiple Sclerosis patients between Stratus and Cirrus OCT. For calcutlation of sensitivity, a reference group of age and sex matched healthy controls was used (control eyes n=58 with 58 Stratus and 18 Cirrus OCT measurements).

	Stratus OCT				Cirrus OCT				Pearson		paired t
RNFLT	**Mean [µm]**	**SD [µm]**	**Sens.**	**Spez.**	**Mean [µm]**	**SD [µm]**	**Sens.**	**Spez.**	**Corr.**	***p***	***p***
Average	**96,98**	17,84	19,6%	98,0%	**87,48**	14,15	27,1%	100,0%	0,939	<0,001	<0,001
Quadrants:											
I	**123,60**	24,98	14,9%	100,0%	**115,78**	22,85	18,4%	100,0%	0,935	<0,001	<0,001
S	**114,98**	20,28	14,3%	94,4%	**105,48**	19,06	24,0%	93,8%	0,843	<0,001	<0,001
N	**80,24**	20,07	10,9%	98,2%	**69,56**	11,69	14,6%	93,3%	0,729	<0,001	<0,001
T	**68,76**	20,28	19,1%	97,8%	**58,02**	15,53	21,3%	100,0%	0,894	<0,001	<0,001
Clock Hours:											
SN	**104,34**	24,02	12,2%	96,4%	**93,34**	23,24	27,7%	86,7%	0,752	<0,001	<0,001
S12	**109,54**	25,28	16,0%	91,1%	**102,22**	24,66	18,0%	100,0%	0,805	<0,001	<0,001
ST	**131,20**	24,51	4,3%	98,0%	**122,50**	24,31	19,1%	100,0%	0,890	<0,001	<0,001
TS	**81,18**	24,10	13,0%	98,0%	**69,74**	21,09	26,5%	93,3%	0,856	<0,001	<0,001
T12	**52,28**	15,27	17,8%	100,0%	**45,62**	10,98	22,4%	100,0%	0,851	<0,001	<0,001
TI	**72,60**	24,92	18,6%	100,0%	**59,10**	18,21	23,4%	100,0%	0,914	<0,001	<0,001
IT	**135,62**	33,78	16,3%	100,0%	**126,82**	33,80	17,0%	100,0%	0,945	<0,001	<0,001
I12	**130,10**	28,51	2,1%	98,2%	**124,64**	25,26	10,4%	100,0%	0,875	<0,001	0,007
IN	**105,26**	23,99	4,1%	100,0%	**97,24**	20,15	6,1%	100,0%	0,913	<0,001	<0,001
NI	**73,98**	20,42	6,3%	96,4%	**63,72**	12,18	6,0%	92,9%	0,828	<0,001	<0,001
N12	**68,04**	19,35	4,3%	98,1%	**57,70**	9,07	4,3%	100,0%	0,609	<0,001	<0,001
NS	**98,44**	25,78	14,9%	96,5%	**87,34**	20,06	10,9%	100,0%	0,690	<0,001	<0,001

Abbildung 7.29: Referenzwerte zur Bestimmung der Qualitätsverbesserung des eigenen Verfahrens. Die Werte basieren auf der Studie [BBD+09] und zeigen Sensitivitäten bei sehr hoher Spezifität, von einzelnen Attributen, bei Probanden in einer Alters- und Geschlechtsübereinstimmenden Datenbank. Diese Werte liegen von zwei Messgeräten vor und zeigen auch eine Qualitätsverbesserung von Stratus OCT zum Cirrus OCT.

7.4.6 Ergebnis der Datenauswertung

Nach der Durchführung der Auswertung stehen nun die Ergebnisse zur Verfügung, die in Tabelle 7.30 zusammengefasst sind. In der Tabelle kann man sehen, welche Algorithmen besser als andere sind und wie die Attributauswahl die Ergebnisse beeinflusst. In Abbildung 7.31 ist eine andere Art der Ergebnispräsentation zu sehen, diese Auswertung bezieht sich jedoch nur auf den Wert AUC von Ergebnis 1 und Ergebnis 2.

			Ergebnis 1			Ergebnis 2		
Name des Versuchs	Attribute	Verfahren	AUC	95%	98%	AUC	95%	98%
0000	1	default	0,500	00,00	00,00	0,500	00,00	00,00
1000	2	default	0,500	00,00	00,00	0,500	00,00	00,00
2000	3	default	0,500	00,00	00,00	0,500	00,00	00,00
3000	4	default	0,500	00,00	00,00	0,500	00,00	00,00
0001	1	SVM linear	0,737	37,00	33,00	0,817	36,00	36,00
1001	2	SVM linear	0,748	39,00	32,50	0,818	39,00	32,00
2001	3	SVM linear	0,751	40,00	34,00	0,790	43,00	43,00
3001	4	SVM linear	0,756	38,00	32,00	0,773	39,00	34,50
0002	1	Bayes	0,733	39,00	27,50	0,796	39,00	39,00
1002	2	Bayes	0,745	40,00	34,00	0,808	36,00	34,50
2002	3	Bayes	0,745	39,00	37,50	0,770	41,00	39,00
3002	4	Bayes	0,759	41,00	34,00	0,809	45,00	39,00
0003	1	SVM poly	0,702	25,00	08,00	0,703	15,50	02,50
1003	2	SVM poly	0,752	39,00	35,00	0,817	40,50	32,00
2003	3	SVM poly	0,754	39,00	33,00	0,794	45,00	39,00
3003	4	SVM poly	0,753	41,00	34,00	0,775	40,50	36,00

Abbildung 7.30: Tabelle mit den Ergebnissen des DataMining. Ergebnis 1 ist das gemittelte Ergebnis der Bewertung innerhalb der X-Validierung und Ergebnis 2 das Ergebnis nach dem Test auf den Testdaten. In der Spalten 95 % und 98 % stehen die jeweiligen Sensitivitäten, zu den gegebenen Spezifitäten. Der Spaltenname AUC steht für die Fläche unter der ROC-Kurve. In Attribute steht die Auswahl der Attribute und in der Spalte Verfahren der Lernalgorithmus.

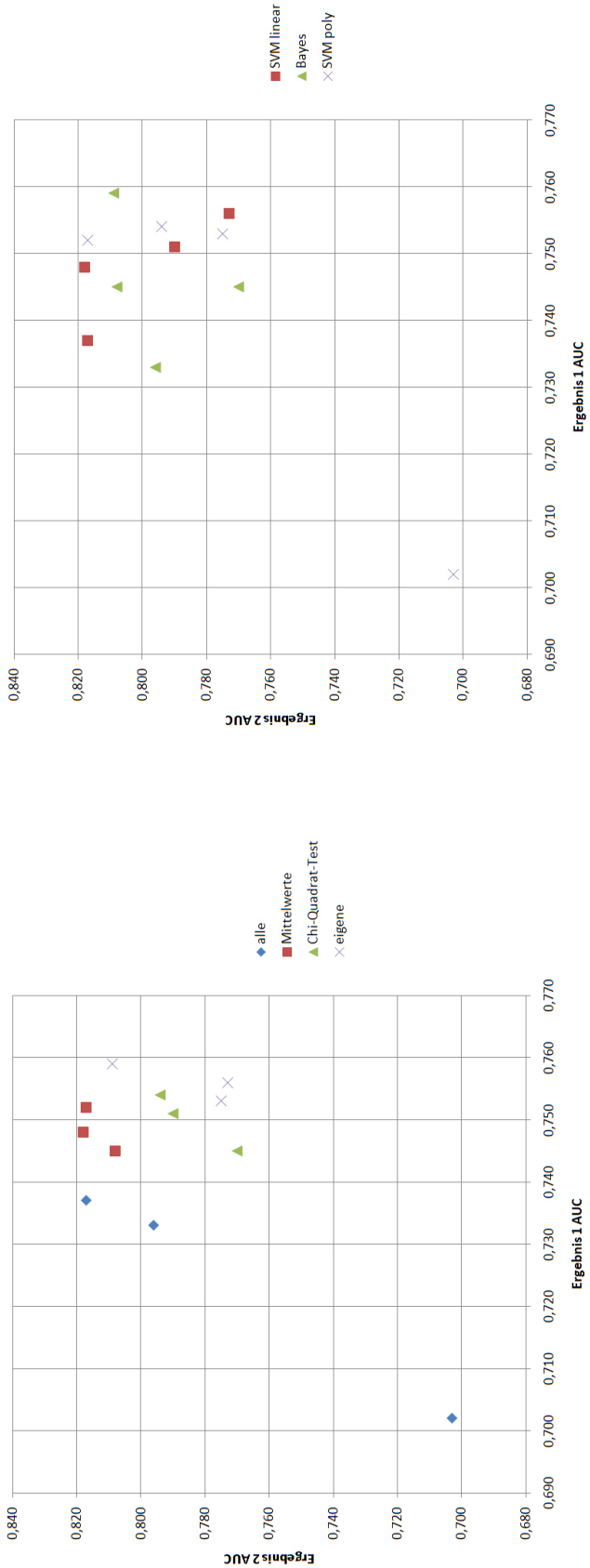


Abbildung 7.31: Ergebnisse der Datenauswertung in Diagrammform, hierbei wird nur der Wert AUC von Ergebnis 1 und Ergebnis 2 ausgewertet. Links sind die Werte nach der Attributauswahl sortiert und rechts nach dem angewendeten Lernalgorithmus.

Platzierung der Ergebnisse

Unter den Lernstrategien haben sich einige besser bewährt als andere, in der Tabelle 7.4 sieht man eine Sortierung der Ansätze. Die Reihenfolge beruht auf dem Ansatz, dass ein Verfahren sowohl in Ergebnis 1 als auch in Ergebnis 2 gut abschneiden muss.
Das nach diesen Kriterien beste Ergebnis erreichte der Naive Bayes Algorithmus, mit der eigenen Attributauswahl, knapp gefolgt von dem Naive Bayes Algorithmus mit der Attributauswahl durch den ChiQuadratTest. Die Sensitivität des Gewinners liegt bei einer Spezifität von 98 % in Ergebnis 1 bei 34 % und in Ergebnis 2 bei 39 %.
Ein anders Kriterium, für die Suche des besten Ergebnisses, ist nur den Wert der Sensitivität bei 98 % Spezifität in Ergebnis 2 zu betrachten. Hier erreichte die SVM mit linearem Kernel und der Attributauswahl nach dem ChiQuadratTest, mit einem Wert von 43 % Sensitivität, das beste Ergebnis. Vergleicht man dieses, mit dem Ergebnis von dem Ansatz, der den gleichen Lernalgorithmus aber nur die Mittelwerte als Attribute verwendet, so erhält man eine Verbesserung von 34, 4 % allein durch die Attributauswahl.

Attributauswahl	**Lernalgorithmus**	**Rang 1**	**Rang 2**	**Average**
eigene	Bayes	2	1	1,5
chi	Bayes	1	5	3
chi	SVM linear	5	2	3,5
eigene	SVM poly	3	6	4,5
chi	SVM poly	7	3	5
average	SVM poly	4	7	5,5
alle	Bayes	11	4	7,5
average	Bayes	6	10	8
average	SVM linear	8	9	8,5
alle	SVM linear	10	8	9
eigene	SVM linear	9	11	10
alle	SVM poly	12	12	12

Tabelle 7.4: Tabelle mit der Auflistung der DataMining-Setups nach dem Platz den sie belegen. Rang 1 ergibt sich aus dem Abschneiden des Setups nach dem Berechnen von (Sensitivität bei 95 % Spezifität + Sensitivität bei 98 % Spezifität) $*$ AUC von Ergebnis 1. Rang 2 ergibt sich aus dem Abschneiden des Setups nach der gleichen Berechnung allerdings von Ergebnis 2.

8 Diskussion

In dieser Arbeit wurde erfolgreich bewiesen, dass durch Anwenden von DataMining eine Verbesserung der Diagnose der MS anhand von RNFLT-Messungen möglich ist. Wie die Ergebnisse einzuschätzen sind, soll in diesem Kapitel diskutiert werden.

8.1 Dateneingabe

8.1.1 Export der Messungen

Der Export ist mit den momentanen Mitteln sehr zeitaufwendig und kompliziert. Aus diesem Grund ist ein neues Gespräch mit der Firma, welche die Software für das OCT-Gerät herstellt, fest geplant. Dort soll versucht werden eine Kooperation zu erwirken. Diese soll dem Ziel dienen einen besseren Zugriff auf die aufgenommenen Daten zu erlangen und somit einen durchgängigen Datenfluss zu realisieren.

Datenhaltung

Die Struktur zum Ablegen der exportierten Messungen ist für diese Aufgabe Ideal. Die Struktur bietet für die automatische Erfassung alle Attribute, die für einen minimalen Datensatz benötigt werden. Außerdem sind genügend Redundanzen vorhanden, um Eingabefehler zu detektieren und dennoch den Export nicht unnötig aufwendiger erscheinen zu lassen.
Gespeichert werden die Messungen nach der Eingabe dann in der Datenbank, die für die jetzige Aufgaben gut strukturiert ist. Auch für die Zukunft sollte die Struktur der Datenbank gut anpassbar sein. Durch die Verwendung von eigenen Tabellen für die einzelnen Messungen bleibt der Überblick erhalten. Jedoch fallen jetzt auch schon einige kleine Unstimmigkeiten auf, die man noch ändern könnte. Zum Beispiel sollte man um Speicherplatz zu sparen, einige IDs in einen GUID Datentyp umwandeln. Es gibt sicher noch weitere kleinere Möglichkeiten zur Optimierung, im Großen und Ganzen hat sich die Struktur der Datenbank jedoch bewährt.

8.1.2 Silverlight Oberfläche

Die Silverlight-Anwendung fand anfangs bei den Benutzern keinen großen Gefallen. Die Ärzte haben im klinischen Alltag keine Zeit, sich durch langwierige Wizards zu klicken, weshalb solch ein flaches Design zwar auf größere Anerkennung stieß, die reine Eingabe generell aber zu lange dauerte. Zur Auswertung und Verwaltung der Messungen kann die Anwendung hingegen gut genutzt werden. Sollte das Problem mit der Datenintegration

vom OCT-Gerät in Zukunft gelöst werden, steht dem erfolgreichen Einsatz der Software nichts entgegen.
Da eine sichere, den Vorschriften entsprechende Webanwendung zu entwickeln, den Zeitrahmen überschritten hätte und es zur Zeit nur an einer Studie getestet wird, wurde sie zunächst nur lokal auf einem Laptop bereitgestellt. Somit kann man die Benutzung sicher eingrenzen und den Missbrauch von Daten verhindern. Deshalb war die Entscheidung eine nicht webfähige Testanwendung zu schreiben sinnvoll. Sollte in Zukunft durch die Verbesserung der Ergebnisse oder die bessere Datenintegration die Nachfrage größer werden, so sollte eine Umstellung auf eine rein webbasierte Anwendung unbedingt erfolgen.
Die Auswertungen führten zu starkem Interesse der Ärzte, weshalb weitere Auswertungen erfolgen sollten. Diese sollten auch mit graphischen Elementen unterlegt werden. Es können auch noch Daten präsentiert und ausgewertet werden, die bis jetzt noch nicht weiter betrachtet wurden. Dies können weitere Teile der Messungen oder gänzlich andere Messungen sein. Im Speziellen ist es eventuell interessant, das Bild des Augenhintergrundes mit anzuzeigen und auszuwerten. Aus diesem Bild lasst sich eine Fehlpositionierung des Messkreises erkennen.

8.1.3 RnfltImport

Das einfach zu bedienende Programm „RnfltImport“, ist nur aus der Not heraus entstanden, da die Nutzer die Eingebefunktion in „IEyeDoc“ nicht nutzen wollten. Es wurde sofort angenommen, da es automatisch einfache Fehler findet und über eine gute Dokumentation dieser verfügt. So können die Fehler im Nachhinein einfach und effektiv behoben werden.
Der nächste Schritt für diese Anwendung sollte es sein, die Funktionalität in die Hauptanwendung „IEyeDoc“ zu integrieren. Mit diesem Schritt kann man die Anwendbarkeit von „IEyeDoc“ erheblich verbessern.

8.2 Datenbereinigung

Die Eingabe in „IEyeDoc“ funktioniert größtenteils mit Auswahlfeldern, so werden die Fehleingaben minimiert. Manche Eingabefelder beinhalten auch eine Überprüfung des Inhaltes, so werden in einigen Feldern nur Zahlen und in anderen nur Text zugelassen. In einer Weiterentwicklung kann man bei falschen Eingaben auch Vorschläge oder Hinweise zur richtigen Eingabe geben. Dies kann zum Beispiel über Tooltips realisiert werden.
Die automatische Eingabe mit „RnfltImport“ verläuft sehr gut, die am häufigsten auftretenden Fehler werden sicher gefunden und nur sehr wenige rutschen durch das Suchmuster. Die Anzeige der Fehler hilft diese zu Finden und zu Beheben. Momentan ist das Auflösen von Fehlern noch recht schwierig, da es keine Vorschläge oder Lösungsansätze für auftretende Fehler gibt. Die Fehler sind zwar meist immer die gleichen, jedoch muss man sich gut auskennen und ein wenig Hintergrundwissen besitzen, um die Fehler richtig auflösen zu können. In einer Weiterentwicklung zur Webanwendung und der damit verbundenen Erweiterung des Benutzerkreises, sollte die Fehlerauflösung expliziter be-

trachtet werden.
Einige Fehler werden trotz großen Aufwandes weiterhin noch nicht erkannt. Man kann aber wahrscheinlich nie alle möglichen Fehler erkennen, solange man Freitexteingaben zulässt und Menschen die Eingabe bedienen, denn Menschen machen Fehler. Gerade bei solch einer monotonen Arbeit wie dem Export der Messungen schleicht sich Routine ein und es entstehen Fehler. Die Einsicht, dass eine manuelle Überprüfung am Schluss notwendig und sinnvoll ist, brachte dann aber sehr gute Ergebnisse. Zum einen wurden noch Fehler von den Betrachtern gefunden und zum anderen musste man feststellen, dass es recht wenige waren. Bei 523 eingegebenen Messreihen konnten nach der manuellen Überprüfung nur drei Fehler gefunden werden.

8.3 Datenerweiterung

Die Ergebnisse zeigen, dass die Datenerweiterung schon in die richtige Richtung geht. Die Ergebnisse mit den Attributmengen, die auch die selbst generierten Attribute enthalten, sind meist besser als jene, die nur die Originalwerte oder Mittelwerte enthalten. Momentan werden die zusätzlichen Attribute von einem extra Programm namens „RnfltAttributeCalculator“ generiert. Die Funktionalität, die dieses Programm bietet, kann auch in „IEyeDoc“ integriert werden, um so die Redundanz in der Entwicklung zu reduzieren.
Durch intensive Gespräche mit den Fachärzten und die Anwendung von Befragungstechniken konnte deren Wissen genutzt werden, um dadurch neue Attribute generieren zu können. Dies sollte auch zukünftig geschehen, denn die Generierung von neuen Attributen ist keineswegs beendet. Ansätze für weitere Attribute gibt es noch genug. In einem Gespräch mit einem Arzt kam die Idee auf, ein Bild vom Augenhintergrund in die Untersuchung mit einzubeziehen. So kann die korrekte Ausrichtung des Messkreises kontrolliert werden.
Die Idee, die Messreihe als Zufallsverteilung aufzufassen und auf dieser Erkenntnis Berechnungen durchzuführen, führte bis jetzt zu keiner Verbesserung. Es entstanden zwar Attribute, die jedoch keine Verbesserung der Modelle nach sich zogen. Von solchen Rückschlägen darf man sich aber nicht aufhalten lassen. Eine Erkenntnis ist daraus dennoch gewonnen worden: Die Messreihe als Zufallsverteilung aufzufassen, bring keinen ersichtlichen Mehrwert.
Ein anderer Ansatz, der noch nicht verfolgt wurde, ist ganz am Anfang der Kette zu finden. Es liegen drei Messungen vor, die zur Zeit gemittelt und ausgewertet werden. Die Berechnung des Mittelwertes verschlingt aber schon Informationen, die noch anderweitig interessant sein könnten. Man sollte auf jeden Fall noch die drei Messungen einzeln untersuchen, ob sich daraus Erkenntnisse erzielen lassen. Auch eine Anwendung einer Mittelwertberechnung mit einem anderen Verfahren, zum Beispiel der Median-Berechnung, kann sinnvoller sein als das jetzige Verfahren. Diese beiden Ansätze sind in einer Weiterentwicklung dringend zu untersuchen.

8.4 Datenauswertung

Bei der Datenauswertung wurde sich absichtlich nur auf ein paar wenige Algorithmen beschränkt, die Ergebnisse sollen einen Anfang liefern solche Verfahren in der Medizin zu etablieren.
Es konnten nur wenige Lernansätze getestet werden, dennoch sind hier schon große Unterschiede zu erkennen. Viele der sehr komplexen Verfahren, die zum Beispiel zu Beginn eine Selektion der Attribute durchführen, zeigen sich als sehr spezialisierte Klassifizierer. Sie können auf ihren Lerndaten sehr gute Ergebnisse erzielen. Wendet man sie auf unbekannte Daten an, so sind die Ergebnisse erheblich schlechter. Die Vermutung liegt nahe, dass die erlernten Modelle zu spezialisiert sind. In diesem Fall spricht man von „Overfitting“. Um dieses zu verhindern, sollte die Auswirkung der Parameterveränderung dieser Lernalgorithmen näher untersucht werden.
Trivialere Ansätze erzielen auf den Lerndaten, durch eine geringere Spezialisierung, schwächere Ergebnisse. Auf unbekannten Daten können sie jedoch ihre Ergebnisse bestätigen. Diese Eigenschaft qualifiziert diese Art von Lernalgorithmen auch für den zukünftigen Einsatz. Ein Beispiel dafür ist der Ansatz, der den Naive Bayes Algorithmus, mit der eigenen Attributauswahl benutzt. Dieser erreicht eine Sensitivität von 34 % bei Ergebnis 1 und 39 % bei Ergebnis 2, bei gegebener Spezifität von 98 %.
Ein wesentliches Ergebnis dieser Arbeit ist die Erkenntnis, nicht alle zur Verfügung stehenden Attribute zum Erlernen eines Modells zu verwenden. Die Auswahl der Attribute spielt somit eine große Rolle. Die Gewichtung nach dem ChiQuadratTest lieferte dort sehr gute Ergebnisse. Das schon im Vorhinein bekannte Attribut des Mittelwertes landete in diesem Test glatt auf Platz eins. Es scheint also gut zur Unterscheidung der Klassen geeignet zu sein.
Die Frage ist nun, geht es noch besser? Und die Antwort darauf muss klar JA heißen. Nahezu alle Ansätze lieferten schon mit den Standardeinstellungen bessere Ergebnisse als die Unterscheidung an Hand des Mittelwertes. Eine Kombination aus dem Wissen über die einzelnen Attribute und dem Wissen über die Lernalgorithmen brachte immer bessere Klassifizierer hervor.
Bei einer Weiterführung der Untersuchungen sollte versucht werden, die Menge der Attribute weiter auf die Relevantesten einzuschränken. Dabei darf nicht vergessen werden, dass das nächste Attribut in der Reihe der Relevantesten, nicht immer auch in der Kombination eine Verbesserung bewirken muss. Manchmal liefert auch eine Kombination von vielen irrelevanten Attributen ein besseres Ergebnis.
Es sollten auch noch viele andere Lernalgorithmen getestet werden. Hier ist gerade erst einmal der Anfang gemacht. Dies soll nicht heißen, dass die hier untersuchten Lernalgorithmen schon ausgereizt sind. Bei diesen gibt es auch noch viele Einstellungen, die noch nicht getestet werden konnten. Als Beispiel soll das Lernen mittels einer Support-Vektor-Machine dienen. Hierbei wurde bis jetzt nur die Einstellung des Kernels und die der Anzahl der Polynome verändert. Beim Kernel wurde auch nur der lineare und polynominale Ansatz getestet. Es stehen noch viel mehr Möglichkeiten zur Auswahl, diese bieten wiederum weitere Einstellungen. Hier ist noch viel Potenzial vorhanden, die geeignetsten Einstellungen für diese Daten zu finden.

Bis jetzt wurden nur die Messungen der Erstuntersuchung ausgewertet. Der ursprüngliche Gedanke war es jedoch eine Verlaufsdiagnose zu ermöglichen. Hierzu sollten dringend die Messungen der Folgeuntersuchungen mit ausgewertet werden. Eine solche Untersuchung hätte in dieser Arbeit zu weit geführt und den Zeitrahmen gesprengt.

8.5 Ausblick für den klinischen Einsatz

Die Aufnahme einer solch großen Menge von Probanden nimmt einige Zeit in Anspruch. Die Technik der OCT ist im Vergleich zu anderen Techniken in der Medizin noch eine recht junge Technologie, weshalb sie ständig weiterentwickelt wird. Die Messungen der Studie, auf der diese Arbeit beruht, muss um einen sinnvollen Vergleich anstellen zu können, immer mit der gleichen Art von Geräten durchgeführt werden. Zur Zeit sind schon Nachfolger der Geräte im Einsatz. Diese können eine detailliertere und wesentlich störungsfreiere Aufnahme realisieren. Eine weitere Verbesserung der Ergebnisse durch die neuen Messgeräte ist deshalb zu erwarten. Vgl.[BBD+09]
Man sollte sich deshalb bewusst sein, dass dies nur ein Anfang ist. Für den klinischen Einsatz in der Medizin reichen die Ergebnisse noch nicht aus. Dennoch verbessert die Auswertung mittels Verfahren der Mathematik und Informatik die Ergebnisse momentan schon um rund ein Drittel. Mit fortschreitender Verbesserung der Messgeräte und der Auswertung der Messungen, kann dieses Verfahren in der Zukunft das MRT in der Diagnose der MS ergänzen. Somit kann dies die Grundlage für eine preiswerte und massentaugliche Verlaufsdiagnose der MS werden.

Abbildungsverzeichnis

Literaturverzeichnis

[BB09] Brandt, Alexander U. ; Bischoff, Sebastian: *Studie IEyeDoc : RNFLT Diagnostic in Multiple Sclerosis and Glaucoma*. 2009. – Unveröffentlicht, liegt der Arbeit bei. Dateiname: „SOP Studie IEyeDoc.pdf“

[BBD+09] Bock, Markus ; Brandt, Alexander U. ; Dörr, Jan ; Pfueller, Caspar F. ; Ohlraun, Stephanie ; Zipp, Frauke ; Paul, Friedemann: *Time Domain and Spectral Domain Optical Coherence Tomography in Multiple Sclerosis : a Comparative Cross Sectional Study*. 2009. – Noch Unveröffentlicht, liegt der Arbeit bei. Dateiname: „Bock et al Stratus Cirrus 07 09 09 AB.pdf“

[BGG96] Bleymüller, Josef ; Gehlert, Günther ; Gülicher, Herbert: *Statistik für Wirtschaftswissenschaftler*. 10. München : Vahlen, 1996

[BHPT99] Beyer, Otfried ; Hackel, Horst ; Pieper, Volkmar ; Tiedge, Jürgen: *Wahrscheinlichkeitsrechnung und mathematische Statistik*. 8. Stuttgart ; Leipzig : Teubner, 1999

[BKNT09] Baun, Christian ; Kunze, Marcel ; Nimis, Jens ; Tai, Stefan: *Cloud Computing : Web-basierte dynamische IT-Services*. Berlin : Springer, 2009

[Bor89] Bortz, Jürgen: *Statistik für Sozialwissenschaftler*. 3. Berlin ; Heidelberg ; Tokio et. al. : Springer, 1989

[Ert09] Ertel, Wolfgang: *Grundkurs Künstliche Intelligenz : Eine praxisorientierte Einführung*. 2. Wiesbaden : Vieweg+Teubner, 2009

[FCZ+06] Frohman, Elliot ; Costello, Fiona ; Zivadinov, Robert ; Stuve, Olaf ; Conger, Amy ; Winslow, Heather ; Trip, Anand ; Frohman, Teresa ; Balcer, Laura: Optical coherence tomography in multiple sclerosis. In: *Lancet Neurol* 5 (2006), S. 853–863

[FFF+08] Frohman, Elliot M. ; Fujimoto, James G. ; Frohman, Teresa C. ; Calabresi, Peter A. ; Cutter, Gary ; Balcer, Laura J.: Optical coherence tomography: a window into the mechanisms of multiple sclerosis. In: *nature clinical practice NEUROLOGY* 4 (2008), S. 12

[GMW03] Gleixner, Christiane ; Müller, Markus ; Wirth, Steffen-Boris: *Neurologie und Psychiatrie für Studium und Praxis*. 3. Breisach : Med. Verl.- und Informationsdienste, 2002/03

[Gol06] Gold, R.: Akute demyelinisierende Erkrankungen des Zentralnervensystems. In: *Neurologie 2006 (Aktuelle Neurologie - Sonderband - Deutsche Gesellschaft für Neurologie)* Sonderband (2006), S. 136–139

[GSK07] Gola, P. ; Schomerus, R. ; Klug, C.: *BDSG - Bundesdatenschutzgesetz Kommentar.* München : Beck, 2007

[Hab95] Haberäcker, Peter: *Praxis der Digitalen Bildverarbeitung und Mustererkennung.* München ; Wien : Hanser, 1995

[Hüb03] Hübner, Gerhard: *Stochastik : eine anwendungsorientierte Einführung für Informatiker, Ingenieure und Mathematiker.* 4. Braunschweig ; Wiesbaden : Vieweg, 2003

[Hem06] Hemmer, B.: Immunpathogenese und Immuntherapie der Multiplen Sklerose. In: *Neurologie 2006 (Aktuelle Neurologie - Sonderband - Deutsche Gesellschaft für Neurologie)* Sonderband (2006), S. 113–115

[Hof06] Hoffmeister, H.: *Impulsfortleitung an der Nervenzelle.* `http://de.wikipedia.org/w/index.php?title=Datei:Impulsfortleitung_an_der_Nervenzelle.png`. Version: 2006. – Stand: 27.09.2009

[JG08] Joël Gubler, Jakov: *Anatomie des Auges.* `http://de.wikipedia.org/w/index.php?title=Datei:Eye_scheme.svg`. Version: 2008. – Stand: 26.09.2009

[Joh92] Johanneson, Rolf: *Informationstheorie.* Bonn ; München : Addison-Wesley, 1992

[Kli01] Klingen, Bruno: *Fouriertransformation für Ingenieur- und Naturwissenschaften.* Berlin ; Heidelberg ; Tokio et. al. : Springer, 2001

[Med03] Meditec, Carl Z.: *Stratus OCT Benutzerhandbuch : Wirklich einfach, wirklich effectiv.* `http://www.zeiss.de/88256DE40004A9B4/0/29B1D53F0547320EC12574EE007535C6/$file/stratusoct_de.pdf`. Version: 2003. – Stand: 09.10.2009

[Nie02] Niemeier, Wolfgang: *Ausgleichungsrechnung : Eine Einführung für Studierende und Praktiker des Vermessungs- und Geoinformationswesens.* Berlin ; New York : de Gruyter, 2002

[Pet99] Petrahn, Günter: *Taschenbuch Vermessung : Grundlagen der Vermessungstechnik.* Berlin : Cornelsen, 1999

[PRE+05] Polman, C. H. ; Reingold, S. C. ; Edan, G. ; Filippi, M. ; Hartung, H. ; Kappos, L. ; Lublin, F. D. ; Metz, L. M. ; McFarland, H. F. ; O'Connor, P. W. ; Sandberg-Wollheim, M. ; Thompson, A. J. ; Weinshenker, B. G. ; Wolinsky, J. S.: Diagnostic Criteria for Multiple

Sclerosis : 2005 Revisions to the „McDonald Criteria“. In: *Ann Neurol* 58 (2005), S. 840–846

[Rüe09] RÜEGG, Christoph: *Math.NET Project*. `http://www.mathdotnet.com/`. Version: 2009. – Stand: 27.10.2009

[RW08] REY, Günter D. ; WENDER, Karl F.: *Neuronale Netze : eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. Bern : Huber, 2008

[SC08] STEINWART, Ingo ; CHRISTMANN, Andreas: *Support Vector Machines*. New York : Springer, 2008

[Sch09] SCHULZE, Alexander: *Rich-Internet-Applikationen : best practices vom Core bis zum Desktop*. München : Entwickler.press, 2009

[WF01] WITTEN, I. H. ; FRANK, E.: *Data Mining - Praktische Werkzeuge und Techniken für das maschinelle Lernen*. München ; Wien : Hanser, 2001

[WR08] WERNER, Michael ; RIEGER, Boris: *Interaktive Webanwendungen mit Microsoft Silverlight 2.0 entwickeln : Das große Buch*. Düsseldorf : Data-Becker, 2008

[Zöf03] ZÖFEL, Peter: *Statistik für Wirtschaftswissenschaftler im Klartext*. München ; Boston et. al. : Pearson Studium, 2003