OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG
INFORMATIK UND MATHEMATIK

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG
REGENSBURG MEDICAL IMAGE COMPUTING

Graduierung von Fazialisparesen durch Methoden des Maschinellen Lernens

Bachelorarbeit
von

Raphael Baumann
Matrikelnummer: 3131486

Fakultät Informatik und Mathematik
Ostbayerische Technische Hochschule Regensburg
(OTH Regensburg)

Gutachter: Prof. Dr. Christoph Palm
Zweitgutachter: Prof. Dr. Brijnesh Jain

Abgabedatum: 5. März 2022

Herr
Raphael Baumann
Am Nordheim 14
93057 Regensburg

Studiengang: Technische Informatik

1. Mir ist bekannt, dass dieses Exemplar der Bachelorarbeit als Prüfungsleistung in das Eigentum des Freistaates Bayern übergeht.

2. Ich erkläre hiermit, dass ich diese Bachelorarbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtlich und sinngemäße Zitate als solche gekennzeichnet habe.

Regensburg, den 5. März 2022

Raphael Baumann

Inhaltsverzeichnis

1 Einleitung

Im Rahmen dieser Bachelorarbeit soll die Forschungsfrage geklärt werden, ob und mit welcher Umsetzung eine Graduierung von Fazialisparesen durch Anwendung von Methoden des Maschinellen Lernens durchführbar ist. Für die Graduierung wird die House-Brackmann Skala angewendet, die als Standartmittel zur Einteilung gilt. Das Thema ist von hoher Relevanz. Zum jetzigen Zeitpunkt existiert kein funktionsfähiges System, das den Grad der Fazialisparese aus gegebenen Bildern unabhängig und korrekt, im Praxisalltag duch das Nutzen von Neuronalen Netzen, feststellen kann.

Als Datensätze für den empirischen Teil der Arbeit stehen Patientenbilder zur Verfügung, die jeweils neun Bilder mit unterschiedlichen Posen beinhalten. Diese wurden vom Universitätsklinikum Regensburg bereitgestellt. Zum Klassenausgleich des Datensatzes wird Oversampling genutzt, welches eine Gleichverteilung aus den bereitgestellten Datensätzen erzeugen soll. Das Verhalten der Genauigkeit mit und ohne Oversampling soll kritisch bewertet werden, ob so bessere Ergebnisse erzielt werden können.

Zwei verschiedene Ansätze werden vorgestellt, anhand derer eine Graduierung durchgeführt werden kann. Dazu werden die verschiedenen Faktoren der House-Brackmann Skala als einzelne Module betrachtet, die in ihrem Bereich als Expertensysteme gelten. Diese Module enthalten jeweils Neuronale Netze, die wiederum die Klassifikation innerhalb der Module und ihren zugeordneten Klassen durchführen sollen. Unter Benutzung von Markerpunkten werden die neun Bilder in Teilregionen zerlegt, die als Eingabe für die Module dienen. Durch das Verwenden eines Automaten oder der Zeilensummenoperation werden die einzelnen Klassen der vier Module fusioniert, um den Ausgabegrad nach House-Brackmann zu bestimmen. Eine weitere Möglichkeit wird betrachtet, welche der Fazialisparese unter der Anwendung der House-Brackmann Skala direkt den Grad von I bis VI zuzuordnen versucht, ohne den Umweg über die Module zu nehmen. Dies dient als Referenz zum Vergleich zwischen Direkt und Modulform.

Verschiedene Verfahren zur Abarbeitung der neun Bilder der Datensätze von Patient*in werden vorgestellt. Im sequenziellen Verfahren werden die neun Bilder der/die Patient*innen hintereinander in die Neuronalen Netze eingegeben. Mit der Anwendung von Early Fusion werden diese vorab konkateniert und als gemeinsames Paket in die Neuronalen Netze eingespeist. Bei der Late Fusion hingegen wird jedem Bild für jedes Modul ein eigenes Netz zugewiesen. Experimentell sollen die verschiedenen Verfahren mit und ohne Oversampling zum Klassenausgleich und jeweils mit beiden Ansätzen der Graduierung nach House-Brackmann durchgeführt werden. Die verschiedenen Experimente werden im späteren Verlauf kritisch bewertet und verglichen.

Auch wird kurz angerissen, in welcher Form Caching zur Beschleunigung der Trainingsphase der Neuronalen Netze zur Anwendung kommen kann. Dazu werden eine externe Datenbank, deren ihren Speicher sich auf einer Festplatte (SSD, HDD) und die im Sourcecode verwendete Datenstruktur Least Recently Used Cache verwendet,

die intern, die benötigten Daten im RAM des Systems, aufbewahrt. Der Beschleunigungsfaktor soll experimental analysiert werden und welche Vor- und Nachteile die Cachingmethoden besitzen.

Zuletzt werden die Vor- und Nachteile der verschiedenen Methoden und Verfahren kurz betrachtet, kritisch bewertet, mit anderen, bereits existierenden Systemen verglichen und eingeordnet. Des Weiteren wird ein kurzer Ausblick auf die Möglichkeiten zur Weiterentwicklung und Anwendung der entstandenen Experimente gegeben. Dazu zählen Verschlüsselung der Daten, einen kurzen Einblick der Vor- und Nachteile der Architekturaufbauten Thick- und Thin-Client und die Informationsfusion mit einer weiteren Skala.

2 Grundlagen

2.1 Fazialisparese und House-Brackmann Skala

Fazialisparese Bei einer Fazialisparese (eng. Facial nerve Paralysis) handelt es sich um eine Funktionsstörung sowohl der Hirnnerven als auch der mimischen Gesichtsmuskulatur. Durch diese Nervenbahnstörung kann eine partielle oder auch vollständige Beeinträchtigung der Muskeln im Gesicht resultieren. Dabei werden vor allem der Lidschluss, Stirnbewegungen (Runzeln und Augenbrauen heben), Mundwinkel und Lippenschluss als offensichtliche Merkmale negativ beeinflusst. Darüber hinaus können ebenfalls Geschmack, Hörsinn, Speichel und Tränenfluss beeinträchtigt sein [1][2].

Die Ursachen für die Entwicklung einer Fazialisparese sind dabei ebenso fassettenreich wie deren Ausprägung:

- Infektionen (z. B. Borrelien)
- Entzündungen
- Traumatische oder geburtstraumatische Schädigung
- Tumore
- Angeborener Defekt
- Idiopathisch (Bell-Parese)
- Verletzung von Nerven

Mit Hilfe verschiedene Skalen kann eine Einordung des Schwereverlaufs einer vorhandenen Fazialisparese erfolgen. Die bekanntesten Skalen sind dabei Sunnybrook und die House-Brackmann Skala. Im Rahmen der Bachelorarbeit wird allein die House-Brackmann Skala zur Anwendung kommen. Diese ist im Vergleich zu anderen Skalen einfacher im Umgang, im Zusammenhang mit der Abstufung und Feststellung der einzelnen Grade und der späteren Datenausgabe an den behandelnden Arzt*in oder direkt an die Patient*innen. Auch ist diese Skala aus medizinischer Sicht ein Standard zur Bewertung der Schwereverlaufes der Fazialisparese [3].

House-Brackmann Skala Die House-Brackmann Skala findet für die Ermittlung des Schweregrades der Fazialisparese Anwendung. Diese Skalierungsmethode wurde von John W. House und Derald E. Brackmann 1983 entwickelt. Das System beinhaltet eine 6-Punkte Skala von römisch I-Normalzustand bis VI-vollständig ausgeprägte Parese (siehe Tabelle 1). Die Einstufung erfolgt, indem der/die Patient*in aufgefordert wird, bestimmte Bewegungen auszuführen. Diese werden durch eine*n Facharzt*in klinisch beobachtet. Daraus resultiert eine subjektive Beurteilung der Schwere der vorliegenden Parese [4].

Grad	Beschreibung	Statisch	Dynamisch		
		Symmetrie	Stirn	Lidschluss	Mund
I	Normal	Normal			
II	Leichte Funktionsstörung	Normal	moderate bis gute Funktion	geschlossen, minimale Anstrengung	leichte Asymmetrie
III	Moderate Funktionsstörung	Normal	leicht bis moderate Funktion	geschlossen, maximale Anstrengung	leicht Betroffen, maximale Anstrengung
IV	Mittelschwere Funktionsstörung	Normal	keine	unvollständig	Asymmetrisch, maximale Anstrengung
V	Schwere Funktionsstörung	Asymmetrisch	keine	unvollständig	leichte Bewegung
VI	komplette Parese	keine			

Tabelle 1: Schweregradeinteilung der Fazialisparese nach House-Brackmann von Grad I, keine sichtbaren Auswirkungen der Parese, bis Grad VI, vollständig ausgeprägte Parese [4].

Ein Vorteil dieser Skala besteht durch die einfache Handhabung darin, dass allein anhand einzelner Figuren und Bewegungsabläufe die Beschreibung der Gesichtsfunktion erfolgen kann. Nachteilhaft sind dabei die subjektive Bewertung des/dem behandelten Arzt*in sowie, dass die regionalen Funktionsunterschiede zwischen dem Grad III und V unempfindlicher für Veränderungen sind. Die Subjektivität der Bewertung erschwert dabei zusätzlich eine Vereinheitlichung und die Vergleichbarkeit von Bewertungen durch verschiedene Ärzt*innen.

2.2 Maschinelles Lernen und Neuronale Netze

Maschinelles Lernen ist ein Teilgebiet der Künstlichen Intelligenz und umfasst im Allgemeinen, durch Methoden und Lernprozesse Zusammenhänge in Datensätzen zu erkennen. Darauf basierend sollen Vorhersagen getroffen werden. Durch das Maschinelle Lernen sollen Optimierungsprobleme gelöst werden, welche aus einer Korrelation zwischen Ausgabewerten und der Eingabe bestehen. Durch selbstlernende Algorithmen sollen diese Modelle die Vorhersagegenauigkeit eigenständig anpassen und den Ausgabewert in der Genauigkeit verbessern, ohne dabei den Algorithmus explizit programmieren zu müssen. Diese Lernverfahren können in drei Kategorien aufgeteilt werden [5][6]:

- Überwachtes Lernen
- Unbewachtes Lernen
- Bestärkendes Lernen

Bei dem überwachten Lernverfahren wird das Modell trainiert, um korrekt die Ausgabewerte eines nicht bekannten Datensatzes (Validationsdatensatz) vorhersagen zu können. Das Training wird durch einen Trainingsdatensatz vollzogen. Dazu sind Relatonen bzw. Paare von Eingabe und Ausgabewerte gegeben. Ziel des Maschinellen

Lernens ist, die Genauigkeit dieser Vorhersage zu maximieren und das bekannte Ergebnis zu treffen. Das Gegenteil vom überwachten Lernen ist das unbewachte Verfahren. Dabei ist die Ausgabe nicht bekannt oder es kann keine präzise Aussage darüber getroffen werden. Anhand der Eingabe sollen so Muster und Zusammenhänge herausgefunden werden. Die dritte Kategorie ist das bestärkende Lernen. Jenes beschreibt eine Methode, womit das Modell bestraft oder belohnt wird, um ein Ergebnis zu optimieren [5].

Im Verlauf dieser Arbeit wird nur das **überwachte Lernen** der Modelle betrachtet, da als Datensatz für die Eingabe Bildmaterial verwendet wird und die erwartete Ausgabe den Grad nach House-Brackmann darstellt. Außerdem ist beim überwachten Lernen die Erfolgswahrscheinlichkeit größer als bei den anderen Verfahren.

Neuronale Netze werden vor allem für die Klassifikation von Daten und Regression eingesetzt. Hierbei werden Parameter des Netzes optimiert, die zwischen den einzelnen Schichten (eng. Layer) liegen. Nicht-lineare Aktivierungsfunktionen transformieren Eingabewerte in andersliegende Wertebereiche. Ergebnisse dieser Funktion dienen in weiterführenden Schichten als Eingabe (siehe Abb. 1). Diese Schichten können auch eine diskrete Faltung oder eine Normalisierung dieser Schichten darstellen. In Kombination mit der Verschachtelung der verschiedenen Layerarten können so beliebig komplexe Modelle gebildet werden [7].

Das Training dieser Netze basiert auf Vorwärts- und Rückwärtsrechnen der Parameter an den Kanten und Knoten der Layer. Dabei läuft zunächst das Eingabebild alle Schichten des Neuronalen Netzes ab, welche Reihen von verschiedenen Transformationen ausführen. Das Ergebnis nach dem Durchlauf wird mit dem realen, zu erwartenden Ergebnis verglichen. Dieses ist innerhalb der Trainingsphase bekannt. Die Abweichung von diesem Vergleich, auch bekannt als Fehler (Loss), wird in der umgekehrten Richtung durch das Netzwerk geschickt. Die Gewichte an den Kanten zwischen den Layern werden anhand des Fehlers angepasst. So soll erreicht werden, dass das Netz sich im Verhalten der Ein- und Ausgabe konvergiert und der Loss abnimmt.

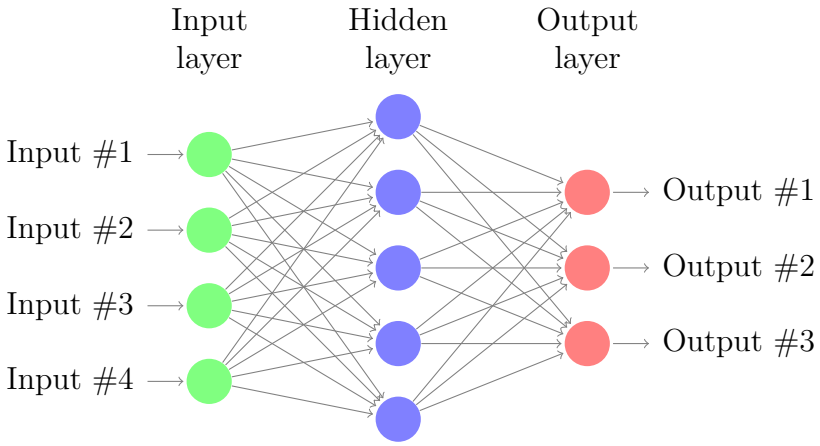


Abbildung 1: Beispielaufbau eines Neuronale Netzes. Die Input Layer repräsentieren die Eingabe in das Netz. Hidden Layer sind versteckte Schichten im Modell. Die Ausgabe auf der linken Seite ist bei einer Klassifikation, Wahrscheinlichkeiten, bezogen auf die zu erwartenden Label. Die Kanten zwischen den Schichten haben Gewichte, anhand derer die Eingabe modifiziert wird (eigene Darstellung).

2.3 Automatentheorie

Automaten stellen einen wichtigen Bestandteile der Informatik dar. In der theoretischen Informatik ist die zu erfüllende Aufgabe im Folgenden so definiert:

> *„Die Aufgabe des Automaten besteht darin, eine Eingabe $w \in \{0,1\}*$ zu durchlaufen (Buchstabe für Buchstabe), und die entsprechenden Transitionen zwischen den Zuständen durchzuführen. Wenn die Eingabe komplett durchlaufen wurde, also keine neuen Buchstaben mehr vorhanden sind, wird das Eingabewort akzeptiert, sofern sich der Automat in einem akzeptierenden Zustand befindet. Anderenfalls wird das Wort abgelehnt. Dies entspricht dem weiter oben eingeführten Wortproblem. Man sagt, der Automat entscheidet die Sprache.“[8]*

Die Eingabe w muss sich nicht zwangsläufig auf den Wertebereich $\{0,1\}*$ beziehen. Aufgabengebiete solcher Maschinen beinhalten Steuerungsaufgaben, Mustererkennung in Daten, Netzwerkprotokolle, Sprachverarbeitung, Klassifizierungen und anderweitige Anwendungsmöglichkeiten. Es gibt zwei zu unterscheidende Hauptkategorien von Automatenklassen, Deterministische Endliche Automaten (DEA) und Nicht-Deterministische Endliche Automaten (NEA).

Ein DEA ist formal komplett durch ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$ definiert. Das 5-Tupel für den DEA enthält folgende Komponenten:

- Q ist eine endliche Zustandsmenge
- Σ ist ein endliches Alphabet
- $\delta : Q \times \Sigma \to Q$ deterministische Übergangsfunktion
- $q_0 \in Q$ ist der Startzustand
- $F \subseteq Q$ Menge aller akzeptierten Endzustände

Ein NEA ist formal durch ein 5-Tupel $M = (Q, \Sigma, \delta, E, F)$ definiert. Der Unterschied zwischen DEA und NEA liegt darin, dass beim NEA mehrere Startzustände vorhanden sein können und die Übergangsfunktionen nicht alle formal definiert sein müssen. Ein NEA kann in einen DEA umgewandelt werden. Das 5-Tupel für den NEA enthält folgende Komponenten:

- Q ist eine endliche Zustandsmenge
- Σ ist ein endliches Alphabet
- $\delta : Q \times \Sigma \to P(Q)$ nicht-deterministische Übergangsfunktion
- $E \subseteq Q$ die Menge der Startzustände
- $F \subseteq Q$ Menge aller akzeptierten Endzustände

2.4 Statistik

Um die verschiedenen Experimente statistisch vergleichen zu können, wird eine Wahrheitsmatrix bzw. Konfusionsmatrix W (eng. Confusion Matrix) erstellt. Diese Matrix ist eine Auflistung aller Ergebnisse eines Trainingsschrittes (Epoch) der Neuronalen Netze. Dazu werden die vorhergesagten Klassen in die Zeile mit der realen Klasse in die Matrix eingetragen (siehe Abb. 2).

Durch die Matrix, sind fünf Metriken bestimmbar. Die erste ist die Sensitivität, welche die Wahrscheinlichkeit angibt, anhand derer ein Objekt richtig klassifiziert wurde. Das Gegenteil ist die Spezifität. Diese gibt an, mit welcher Wahrscheinlichkeit ein Objekt falsch klassifiziert wird. Als weitere Werte lassen sich Spezifität und Segreganz (positiver und negativer Vorhersagewert) berechnen. Diese geben den Anteil der korrekt klassifizierten an der Gesamtheit der klassifizierten Ergebnisse positiv und negativ an. Der F1-Wert ist dabei das harmonische Mittel zwischen Sensitivität und Genauigkeit.

Um für die statistischen Merkmale Sensitivität, Genauigkeit, Spezifität, Segreganz und F1 für die Matrix als Mittelwert berechnen zu können, ist für alle Klassen zunächst notwendig, ihre Einzelwerte zu berechnen. Danach werden alle Klassen summiert und der Mittelwert über ihnen gebildet. Die spezifischen Werte für eine Klasse g, die Wahrheitsmatrix W, die Reihe r und Spalte p können nach den Formeln (1 - 5) ausgerechnet werden [9]:

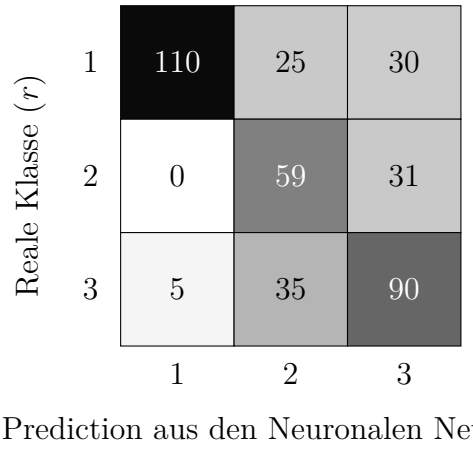


Abbildung 2: Beispiel einer Wahrheitsmatrix W. Das Ergebnis aus den Neuronalen Netzen wird einzeln mit den richtigen Ausgabewert verglichen und in die Zeile mit +1 eingetragen (eigene Darstellung).

$$Sensitivität(TPR) = \frac{TP}{TP + FN} = \frac{W_{g,g}}{\sum_{n=1}^{n} r_g} \tag{1}$$

$$Genauigkeit(PPV) = \frac{TP}{TP + FP} = \frac{W_{g,g}}{\sum_{n=1}^{n} p_g} \tag{2}$$

$$Spezifität(TNR) = \frac{TN}{TN + FP} = \frac{W_{\neg g,\neg g}}{\sum_{n=1}^{n} r_{\neg g}} \tag{3}$$

$$Segreganz(NPV) = \frac{TN}{TN + FP} = \frac{W_{\neg g,\neg g}}{\sum_{n=1}^{n} p_{\neg g}} \tag{4}$$

$$F1 = \frac{2 * PPV * TPR}{PPV + TPR} \tag{5}$$

Die Diagonalpositionen der oben gezeigten Matrix repräsentieren alle Richtig-Positiv Klassifizierungen der respektiven Klasse. Anhand dieser Metriken kann so herausgefunden werden, ob ein Neuronales Netz in eine Über- bzw. Unteranpassung (eng. Over- und Underfitting) hineinläuft. Ein weiteres Indiz für einen guten Lernvorschritt ist, wenn sich der F1-Wert asymptotisch Richtung 1 annähert. In der Szene des Maschinellen Lernens ist der F1-Wert der Maßstab.

3 Stand der Technik

Im folgenden Kapitel soll der momentane Stand der Technik für die zugrundeliegende Aufgabenstellung erläutert werden. Dabei werden zwei verschiedene Herangehensweisen kurz erläutert. In Abschnitt 3.1 wird dabei die Graduierung der Fazialisparese durch die Anwendung von Segmentierung zur Extraktion der wichtigsten Gesichtseigenschaften veranschaulicht. Eine andere Herangehensweise besteht darin, mit Videoframes vergleichsbasiert mit einer aufgenommenen Referenzstellung und anderen Posen den House-Brackmann Grad (H-B Grad) festzustellen (Kapitel 3.2).

3.1 Segmentierung basierte Methode

Eine Methode zur Erkennung einer Fazialisparese ist die Anwendung von Segmentierung, um die Gesichtsmerkmale aus den Bildern zu extrahieren. Diese semantischen Merkmale enthalten räumliche Eigenschaften des/der Patient*in, die für die Klassifikation des Schweregrades aus optischer Sicht unerlässlich sind.

Der Experimentaufbau von T. Wang et al. enthält eine kaskadierende Encoder Netzstruktur zur Ausführung der Bewertung der Fazialisparese (siehe Abb. 3).Auf Basis von Neuronalen Netzen besteht dieser Aufbau aus zwei Komponenten: semantischer Segmentierung von Gesichtsattributen und ein Klassifikationsnetz über eine vorher

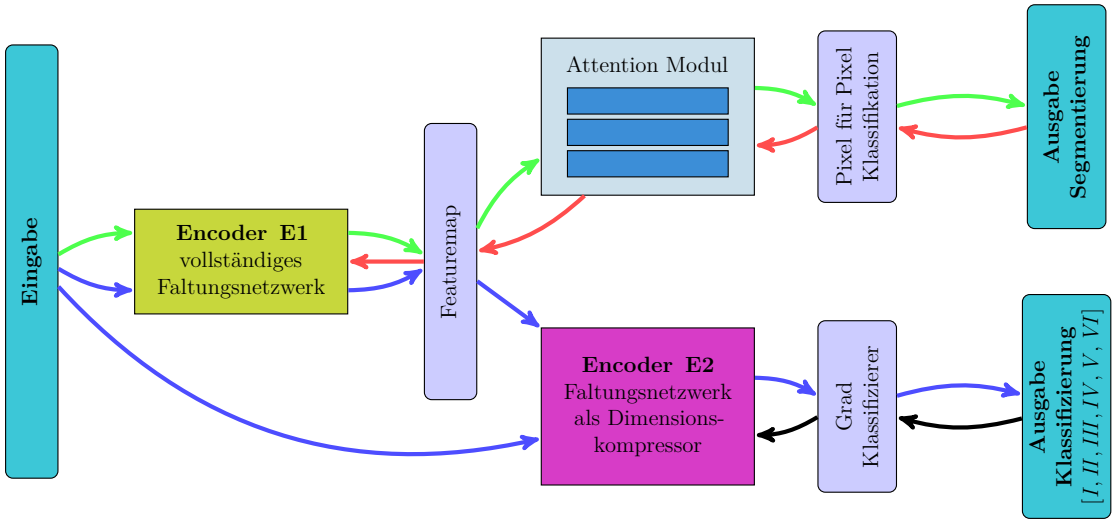


Abbildung 3: Darstellung der Flowchart vom Paper Automatic Facial Paralysis Evaluation Augmented by a Cascaded Encoder Network Structure. Die blauen und grünen Pfeile stellen den Evaluierungsdatenfluss, Rot und Schwarz jeweils den Trainingsfluss dar [10].

extrahierte Funktionskarte (Encoder). Dieser Encoder komprimiert die Bilddaten Pixel für Pixel durch ein Faltungsnetzwerk (eng. Convolution). Verwendet wird dazu ein vortrainiertes ResNet-101 ohne Verkleinerung des Eingabematerials. Als Training für den Encoder diente in der Studie ein Mix aus mit und ohne Fazialsparese, um ihn separat vom Rest vorab zu trainieren. Dadurch konnten ausreichende, räumliche Informationen über die Gesichter gewonnen werden, die für die Segmentierung und die Graduierung benötigt werden. Die daraus enthaltenen Daten aus dem Encoder dienen als Feature für die Segmentierung und die direkte Klassifikation des Grades nach House-Brackmann. Bei der Segmentierung handelt es sich um eine Pixel für Pixel Klassifikation über das gesamte Bild. Die Gesichtspartien können so für die betrachteten Teilregionen der House-Brackmann Skala eingefärbt werden. Als Ausgabe dienen die Klassifizierung und das segmentierte Bild, die zur weiteren, klinischen Einschätzung durch einen Facharzt*in zurückgeliefert werden. Durch die Anwendung des kaskadierenden Trainings wurde so ein F1-Wert von 95.85% erzielt [10].

Der große Vorteil von dieser Methode liegt darin, dass die Segmentierung getrennt von der Graduierung eine Rückgabe liefert. Aus der Segmentierung der Bilder kann so erkannt werden, welche Bereiche des Bildes für die Detektion genutzt wurden und ob es sich um die richtigen Regionen der Gesichtspartien handelt. Auch können Rückschlüsse auf die enthaltene Featuremap gezogen werden, die für den Klassifizierer des Grades auch von Relevanz ist, da Segmentierung und Klassifizierung dieselbe Featuremap benutzen.

3.2 Vergleichsbasierte Erkennung anhand eines Videos

Eine weitere Möglichkeit neben der Segmentierung ist es, aus einem Video herausgeschnittene Frames mit einer Referenz zu vergleichen. Dazu werden fünf unterschiedliche Gesichtsbewegungen durchgeführt: Augenbraun Heben, Augen sanft sowie forciert Schließen, Nase Rümpfen und Lächeln. Die Videosequenz beginnt mit der Referenzansicht, wobei sich der/die Patient*in in Ruheposition befindet. Nach jeder Bewegung kehrt er/sie auch wieder in die Ausgangsposition zurück. Anhand von Framesubstraktionen können die einzelnen Positionen aus dem Video entnommen werden.

Nachdem die einzelnen Posen aus dem Video herausgezogen worden sind, werden die einzelnen Gesichtsregionen lokalisiert. Um die Gesichtskonturen zu detektieren, wird ein Sobel-Filter angewendet. Dieser nutzt eine Faltungsoperation, die aus dem Bild einen Gradienten erzeugt. Die Bereiche mit der größten Intensität, also den Stellen mit den stärksten Helligkeitsänderungen, werden hervorgehoben. Die anderen Bereiche werden durch Grauwerte dargestellt. Alle entscheidenden Features des Gesichtes (Augenbraun, Nase, Mund und Augen) sind dunkler als die normale Hautfarbe. Durch eine Verschiebung der Pixelwerte lassen sich diese hervorheben und deren absolute Position im Bild feststellen.

Anhand dessen werden vier Eingabewerte berechnet, die für eine Klassifikation mit einem Neuronalen Netz benötigt werden (siehe Abb. 4). Dazu zählen die relative Pixelveränderung in den einzelnen Regionen, die sich von der Ruheposition bis zu dem Höhepunkt der dargestellten Bewegung ausrechnen lässt. Auch der Beleuchtungs-

kompensationsfaktor ist von Bedeutung. Dieser misst die Unterschiede zwischen der dysfunktionalen und der normalen Seite, um die Intensität und Ausprägung der Funktionsstörung ermitteln zu können. Die weiteren beiden Eingabewerte sind Faktoren, die sich aus dem optischen Fluss der Regionen darstellen lassen. Dazu werden Vektoren genutzt, die von der Spiegelachse - dem Nasenrücken - und beiden Gesichtshälften in einem Raster die Richtungsänderungen und Falten im Gesicht zeigen sollen. Aus dieser Grafik werden die Symmetrie in vertikaler Richtung sowie die Stärke dieser Vektoren berechnet. Die genannten Faktoren dienen als Eingabe in ein Klassifikationsnetzwerk, das nach der House-Brackmann Skala den Grad der Parese feststellt [11].

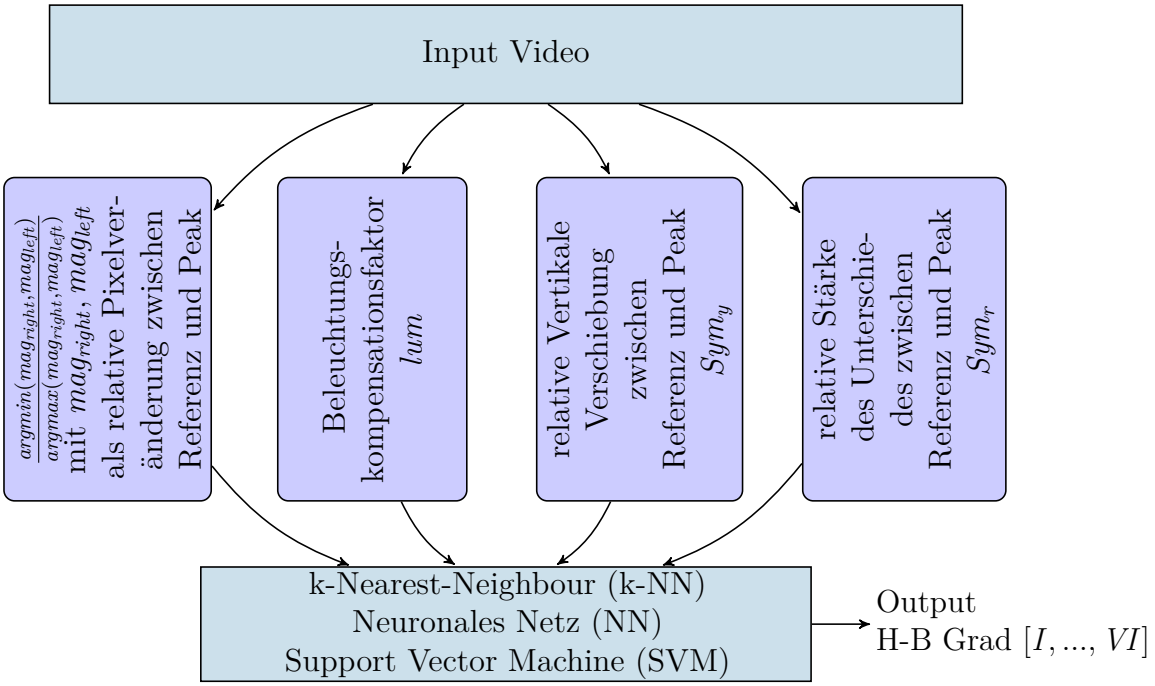


Abbildung 4: Nachgestellte Interpretation der beschriebenen Vorgehensweise für die Klassifikation. Eingabefaktoren für die im Paper verwendeten sind Pixeldichte, Beleuchtungsfaktor, relative Vertikale und Stärke jeweils auf ein Referenzframe und dem Peak im Video betrachtet [11].

4 Material und Methoden

4.1 Material

Als Stichprobe stehen die Datensätze von insgesamt 86 Patient*innen zur Verfügung. Bereitgestellt wurden die Datensätze an Patient*innen vom **Universitätsklinikum Regensburg**. Die dazugehörigen Grade der House-Brackmann Skala sind hierbei innerhalb der Stichprobe allerdings nicht gleichverteilt vorhanden. Der Grad VI ist prozentual ungefähr im gleichen Maße vorhanden wie die Grade I-V zusammen (siehe Abb. 5). Der Grad I ist nur mit eine*r Patient*in vertreten. Diese ungleiche Verteilung wird bei der späteren Auswirkung berücksichtigt werden.

Jeder Datensatz jede*r Patient*in besteht aus neun verschiedenen Bildern, die verschiedene Posen darstellen. Die Gesichtsausdrücke in den Einzelbildern können jeweils einem konkreten Teil des House-Brackmann Scores zugeordnet werden. Zudem werden die normalerweise in Bewegung stattfindenden Ausdrücke statisch im Bild eingefangen.

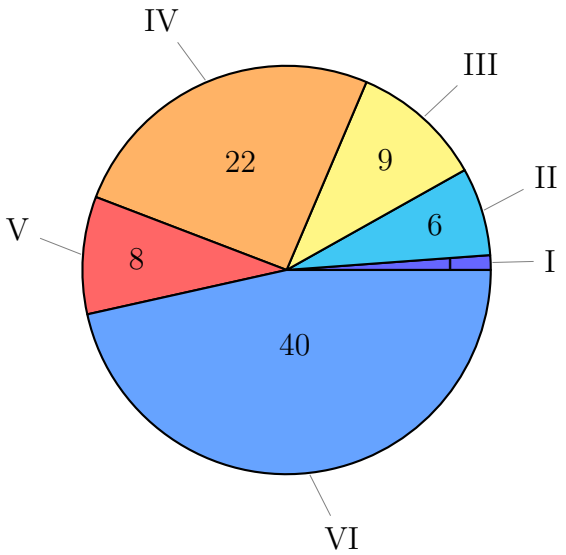


Abbildung 5: Verteilung der einzelnen Grade der House-Brackmann Skala des zu Verfügung stehenden Datensatzes. Klar erkennbar ist das ungleiche Vorhandensein der Grade.

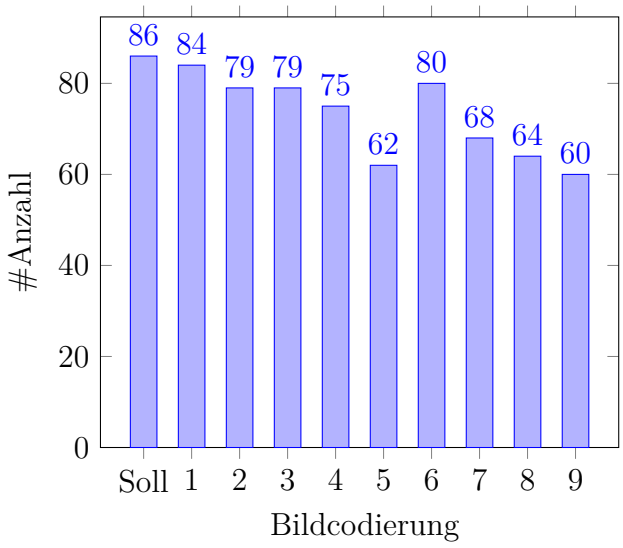


Abbildung 6: Anzahl der vorhandenen Einzelbilder aller Patient*innen. Der Sollwert beträgt 86 für alle Bildcodierungen.

Die Bildcodierung (Posendarstellung) dieser neun Bilder lautet:

1. Ruhender Gesichtsausdruck
2. Augenbrauen Heben
3. Lächeln, geschlossener Mund
4. Lächeln, geöffneter Mund
5. Lippen Schürzen, „Duckface“
6. Augenschluss, leicht
7. Augenschluss, forciert
8. Nase Rümpfen
9. Depression Unterlippe

Das Bild mit der Codierung 1 fokussiert sich dabei auf die Symmetrie in Ruhe. Die zweite Pose 2 bildet die Stirn ab. Dadurch sollen die Auswirkungen sowie die Beweglichkeit der Muskeln rund um die Stirnpartie bildlich festgehalten werden (absichtliche Faltenbildung). Der Lidschluss wird mit den Bildern 6 und 7 statisch eingefangen. Ein nicht geschlossenes Auge wird durch das Weiß des Augapfels für das System erkennbar. Die Gesichtsausdrücke auf den Bildern 3, 4, 5, 8 und 9 zeigen den Mund in unterschiedlichen Positionen. So sollen die Unterschiede zwischen den beiden Gesichtshälften am besten aufgezeigt werden. Die Posen für den Mund sind von außen betrachtet einer der Hauptmerkmale für eine Fazialisparese, welche auch schon für einen Laien erkennbar ist.

Problematisch dabei ist, dass nicht für jede*n Patient*in alle neun korrespondierenden Bilder vorhanden sind. Besonders ist dies bei den Bildcodierungen 5, 7, 8 und 9 der Fall (siehe Abb. 6). Beim Trainieren der Neuronalen Netze ist darauf Rücksicht zu nehmen.

4.2 Methode

4.2.1 Module

Für die Detektion des Grades nach House-Brackmann werden die einzelnen Merkmale der Skala in Module eingeteilt (siehe Abb. 7). Ziel dieser Aufteilung ist es, Expertensysteme für die Teilbereiche der House-Brackmann Skala zu bilden. Diese müssen für die Detektion angepasst werden. Die dynamischen Eigenschaften Stirn, Lidschluss und Mund können so nicht als ein Label interpretiert werden. Die neun Eingabebilder für das System stellen die Posen statisch dar. So können alle Eigenschaften bis auf Lidschluss nach der House-Brackmann Skala auf die Grundlabel „Normal“, „minimale Asymmetrie“, „Asymmetrie“ und „Keine“ aufgeteilt werden. Bei Lidschluss kommen stattdessen nur die zwei Möglichkeiten „vollständig geschlossen“ oder „geöffnet“ infrage (siehe Tabelle 2). Nach dieser Tabelle werden so für die einzelnen Module die passenden Label bestimmt. Die Spalten entsprechen dabei den einzelnen Modulen. Die Überschrift der Spalte bezeichnet das einzelne Modul. Der Inhalt dieser sind die zu ermittelnden Klassen für das Modul numemriert von 0 beginnend.

Grad	Symmetrie	Stirn	Lidschluss	Mund
I	Normal	Normal	Vollständig	Normal
II	Normal	Normal	Vollständig	minimale Asymmetrie
III	Normal	minimale Asymmetrie	Vollständig	minimale Asymmetrie
IV	Normal	Keine	Unvollständig	Asymmetrisch
V	Asymmetrisch	Keine	Unvollständig	Asymmetrisch
VI	Keine	Keine	Unvollständig	Keine

Tabelle 2: Angepasste House-Brackmann Tabelle zur Bestimmung der Label/Klassen aus der Originaltabelle 1 (eigene entwickelte Interpretation).

Anhand von Markerpunkten im Gesicht der Patient*innen werden die neun Bilder jede*r Patient*in in die Bestandteile der Skala, die Regions of Interest, aufgeteilt. Dabei werden nur die relevanten Teile betrachtet, um proaktiv zu verhindern, dass die Neuronalen Netze nicht notwendige Eigenschaften als relevante Merkmale interpretieren. Zu den nicht notwendigen Bestandteilen der Bilder zählen der Hintergrund, welcher verschiedene Farben und Helligkeitsstufen annehmen kann, und der Körper der Patient*innen ab dem Kinn abwärts. Nur der Kopf wird für die Feststellung der Parese benötigt. Die Landmarks, die für die Einteilung in die Regionen verantwortlich sind, müssen sich hierfür immer an derselben Position P im generierten Array befinden (siehe Abb. 8). Jedem Modul können so die relevanten Punkte (siehe Tabelle 3) für ihr Modul sowie das Ausgabearray, welches die neun Bildausschnitte beinhaltet, zugeordnet werden. Für die Berechnung der neuen Eckpunkte für den Bildausschnitt werden die zugeordneten Landmarks P in die nachfolgenden Formeln eingesetzt [12]:

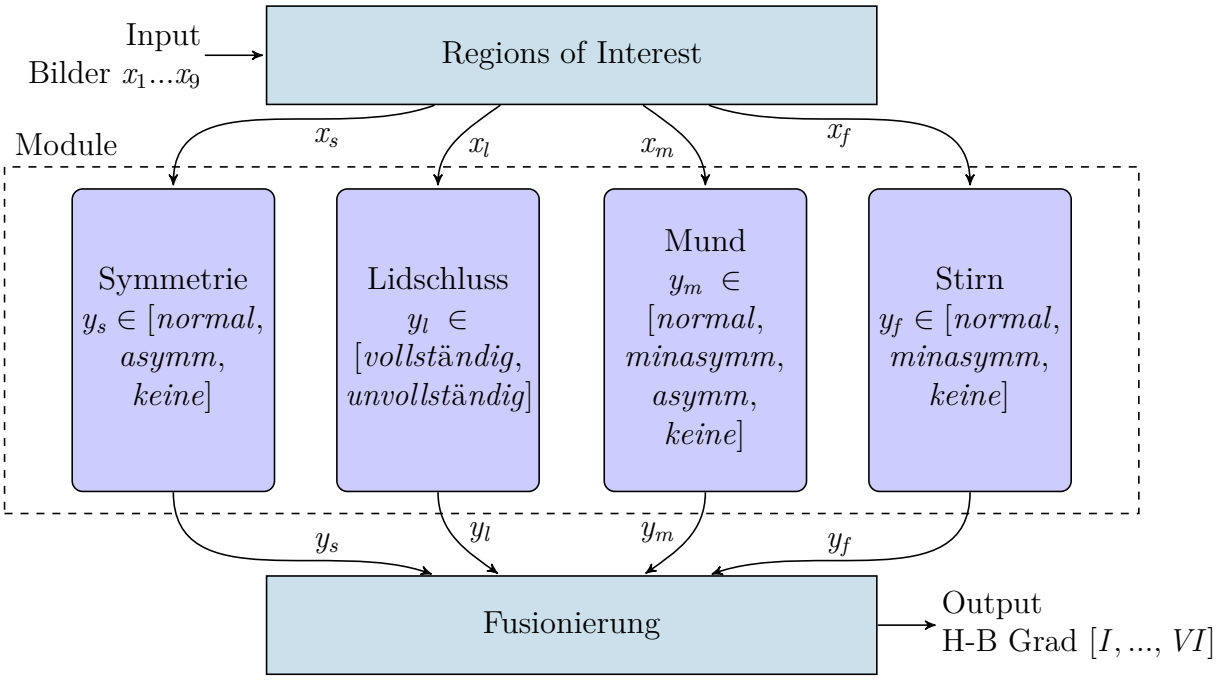


Abbildung 7: Schematische Darstellung der Module. Die Eingabebilder $x_1, ..., x_9$ werden für jedes Bild in die Regions of Interest zerschnitten. Diese wandern als Stack x_s, x_l, x_m, x_f (zerschnittenes Bildmaterial) in die einzelnen vorgesehenen Module. Danach werden diese zum H-B Grad fusioniert.

$$
\begin{aligned}
a_{min} &= min(P[:,0]) \quad | \text{ Extrema links} \\
a_{max} &= max(P[:,0]) \quad | \text{ Extrema rechts} \\
b_{min} &= min(P[:,1]) \quad | \text{ Extrema oben} \\
b_{max} &= max(P[:,1]) \quad | \text{ Extrema unten}
\end{aligned}
\qquad
\begin{aligned}
&\text{Zu erfüllende Bedingungen:} \\
&1. \quad a_{min} < a_{max} \\
&2. \quad b_{min} < b_{max} \\
&3. \quad a_{min}, a_{max}, b_{min}, b_{max} \in \mathbb{N}
\end{aligned}
\tag{6}
$$

Dabei entsprechen (a_{min}, b_{min}), (a_{min}, b_{max}), (a_{max}, b_{min}) und (a_{max}, b_{max}) den vier neuen Ecken des benötigten Bildfragments. Die Ecken spannen ein Rechteck auf, dessen Inhalt als Eingabebild für eines der Module (z. B. Lidschluss nur Punkte $P(36)$ bis $P(47)$) relevant ist. Alles was sich nicht innerhalb des Rechteckes befindet, wird weggeschnitten.

Modul	zugeschnittene Eingabebilder als Stack	Betrachtete Landmarks
Symmetrie	x_s	P(00) - P(68)
Lidschluss	x_l	P(38) - P(47)
Mund	x_m	P(48) - P(68)
Stirn	x_f	P(17) - P(26)

Tabelle 3: Zuordnung der Module zu den relevanten Punkten der Landmarks und die Eingabebilder in die Module nach dem Ausschneiden vom Originalbild x.

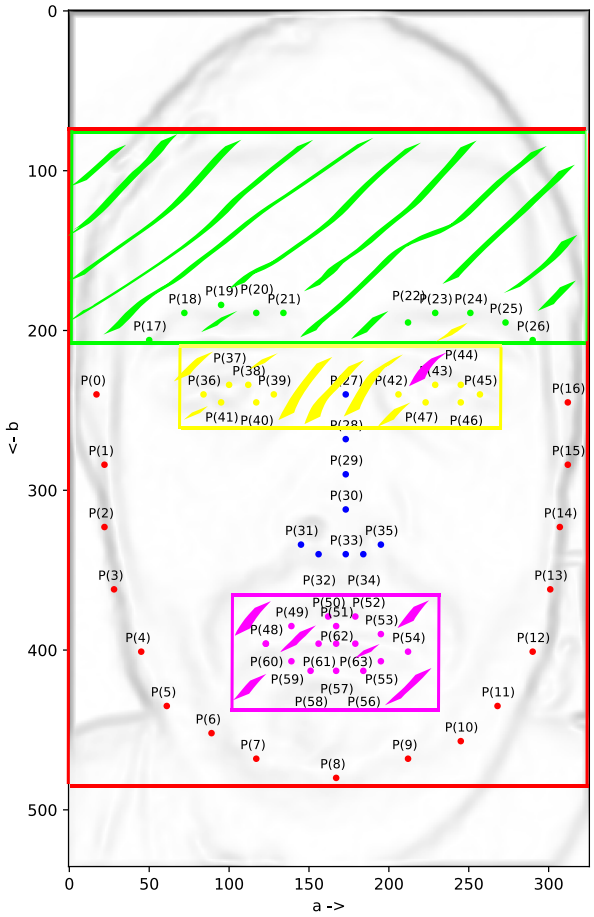


Abbildung 8: Anschauliche Darstellung der Regions of Interest. Inhalt des roten Rahmens ist die Eingabe für das Modul Symmetrie, grün für die Stirn, gelb der Lidschluss und pink für den Mund. Aus Datenschutzgründen durch einen Sobel-Filter verändert.

Ein Offset wird dazu noch in alle Richtungen addiert bzw. subtrahiert, um Ungleichheiten in der Landmarkberechnung auszugleichen, falls die Kontur nicht korrekt getroffen wurde. Wenn neun Bildfragmente für eines der Module berechnet worden sind, werden diese als Stack in die nächsttiefere Schicht weitertransportiert.

Die einzelnen Module Symmetrie, Lidschluss, Mund und Stirn (siehe Abb. 7) sind vortrainierte ResNet18 Netze vom ImageNet Contest. Benutzen von vortrainierten Netzen spart Trainingszeit an den Layern durch die schon vorhandenen angepassent Parametern vom ImageNet Contest. Die einzelnen Gewichte an den Kanten müssen so nur noch verfeinert werden, um das gewünschte Ergebnis zu erzielen. Für alle Experimente (siehe Kapitel 5) ist das gewählte Netz aus Gründen der Vergleichbarkeit der Ergebnisse identisch. Die Abbildungsvorschrift für die Relation Eingabebild x mit der Pixelgröße a, b und drei Schichten für den Farbraum RGB (bei Konkatenation vielfaches von 3) zu auszugebenden Wahrscheinlichkeiten der Klassen y mit der Klassenlänge k ist:

$$y : \mathbb{R}^{3\times a\times b} \rightarrow \mathbb{R}^k, y(x) = resnet18(x) \tag{7}$$

Jedem Eingabebild ist eine natürliche Zahl zugeordnet, die innerhalb der Trainingsphase und der Evaluierungsphase zu ermitteln ist. Diese Zahl ist letztendlich eine Nummer, die ein Label/Klasse nach der oben genannten Tabelle repräsentiert. Nummeriert werden diese von 0 beginnend. Für das Modul Symmetrie und Stirn 0-2, Lidschluss 0 und 1, Mund 0-3. Die Operation (7) liefert Wahrscheinlichkeiten für alle Klassen eines Moduls. Das Ziel ist innerhalb der Phasen die vorgegebene Klasse des Datensatzes zu treffen. Wird die Nummer der Klasse vom Modell getroffen, ist das somit eine korrekte Klassifizierung. Innerhalb der Trainingsphase werden diese Modelle anhand des Losses, die Abweichung zwischen den ermittelten aus dem Modul und der realen Klasse, das Neuronale Netz pro Modul optimiert anhand des Trainingsdatensatzes. Wie gut oder schlecht die Klassifizierung ist, wird anhand eines Validierungsdatensatzes bestimmt, der unabhängig vom Training ist.

Im Anschluss, nachdem alle Klassen der separaten Module, in der realen Anwendung, ausgerechnet wurden, können diese fusioniert werden, um den Grad nach House-Brackmann zu bestimmen. Dabei gibt es zwei Vorgehensweisen, die methodisch zu bestätigen sind (siehe Kapitel 4.4), dass sie den Grad Ordnungsgemäß ermitteln. Dabei können einerseits alle Wahrscheinlichkeiten direkt als Zeilensumme gebildet werden. So kann direkt der sich am höchsten befindlichen Wert der Zeilensumme der Tabelleneinträge den an der linken Seite befindlichen Grad für eine*n Patient*in angenommen werden. Andererseits kann ein Automat hinzugezogen werden. Dazu werden zunächst die Wahrscheinlichkeiten in die jeweils den Modulen zugeordneten Label umgemünzt. Die Position des höchsten Wertes im zurückgelieferten Array des Ausgabewertes der Neuronalen Netze entspricht der Position des Labels nach der Tabellenspalte von der angepassten House-Brackmann Tabelle. Danach durchlaufen alle vier finalen Klassen jedes Moduls den Automaten. Der Endzustand des Automaten repräsentiert den zu ermittelnden Grad nach House-Brackmann.

4.2.2 Direkte Gradermittlung

Zum Vergleich wird ebenfalls nach der House-Brackmann Skala der Grad der Fazialisparese auf dem direkten Weg, ohne die Skala in Module zu zerlegen, ermittelt. So soll festgestellt werden, ob die Zerlegung in Module sinnvoll ist und inwiefern die Ermittlungsgenauigkeit und Präzision sich unterscheiden. Auch soll so eine Richtung

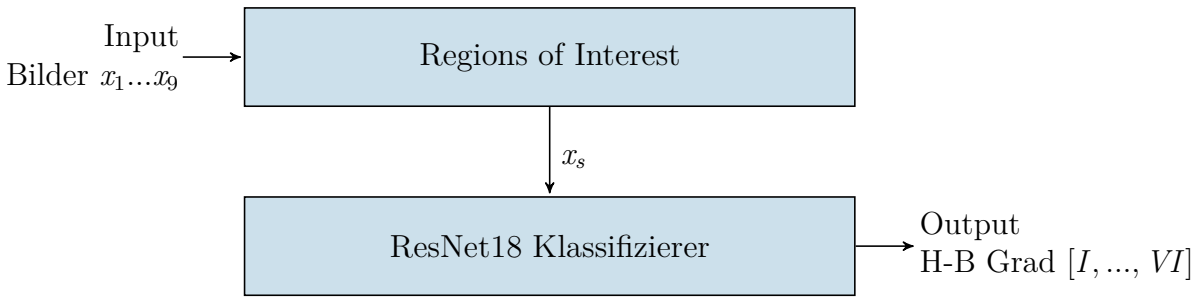


Abbildung 9: Direkte Ermittlung des H-B Grad. Die Bildfragmente x_s die als Eingabe in den Klassifizierer dienen, sind dieselben wie bei der Modulform (siehe 4.2.1) für die Symmetrie.

erkennbar sein, ob sich aus den neun gegebenen Bildern pro Patient*in der Grad ermitteln lässt. Der Aufbau unterscheidet sich minimal von der Modulbauweise. Dabei wird derselbe Stack an Eingabebebildern (x_s) von dem Modul für die Symmetrie genutzt. Diese stellt, wie bekannt, das komplette Gesicht der/die Patient*innen dar. Nur die Klassifikationslabel unterscheiden sich. Anstelle der Klassen für das Modul der Symmetrie $[normal, asymm, keine]$ werden die Label $[I, II, III, IV, V, VI]$ der House-Brackmann Skala als die zu ermittelnde Klassen verwendet (siehe Abb. 9). Eingespart werden alle Berechnungen zu den anderen Modulen und die im Nachhinein stattfindende Fusionierung der Ergebnisse zur Einordnung des Grades nach House-Brackmann.

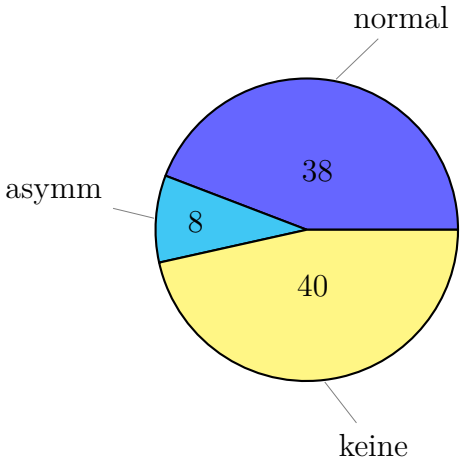
4.2.3 Oversampling zum Klassenausgleich

Die gegebenen Datensätze von Patient*innen mit ihren House-Brackmann Graden ist unausgewogen (siehe Kapitel 4.1). Somit sind auch die Klassen für jedes der Module unterschiedlich verteilt (siehe Abb. 10). Um eine Ausgewogenheit herzustellen, wird das Oversampling angewendet. Hierzu werden die Gewichte ω für jedes Label aus einem Modul ausgerechnet. Allgemein ist Oversampling eine Methode zum Ausgleich von nicht in gleichen Teilen vorhandenen Klassen und ist somit eine Überabtastung eines gegebenen Datensatzes. Dazu wird die Anzahl der Klasse l in den Term (8) hineingegeben. Die einzelnen Gewichte müssen dabei addiert nicht 100% ergeben.

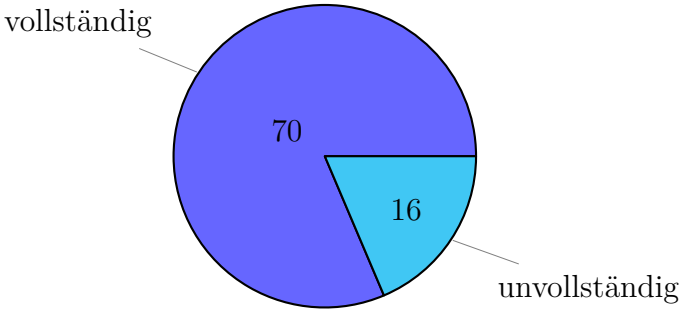
$$\omega = 1/l, \qquad l \in \mathbb{N} \quad \wedge \quad l \neq 0 \tag{8}$$

Ziel ist es, einen ausgewogenen Datensatz zu generieren. Je höher die Anzahl einer Klasse ist, desto niedriger wird das Gewicht ω angesetzt. Ist die Klasse eines spezifischen Modules niedrig, wird ihr ein hohes Gewicht zugeordnet. Durch geschicktes Auswählen aus dem Datensatz durch die zuvor berechneten Gewichte jeder Klasse kann so eine gleiche Verteilung erzeugt werden. Dies erfolgt durch das Duplizieren der höher gewichteten Klassen oder Reduzieren der niedrig gewichteten. Problematisch ist jedoch, dass Bilder von Patient*innen so mehrfach vorkommen können. Das kann sich negativ auf die Trainingsphase der Neuronalen Netze auswirken. Diese können durch die Mehrfachbenutzung falsche Rückschlüsse zur Klassifikationsermittlung ziehen und so das Leistungsvermögen mindern. Oversampling wird ausschließlich auf den Trainingsdatensatz bezogen. Der Evaluierungsdatensatz bleibt unangetastet und wird nicht ausgeglichen.

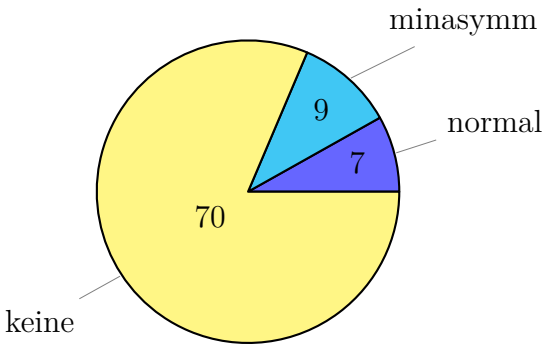
Im Folgenden werden verschiedenste Experimente mit unterschiedlichsten Vorgehensweisen der Datensatzpräparation überprüfen, ob Oversampling ein Vor- oder Nachteil ist.



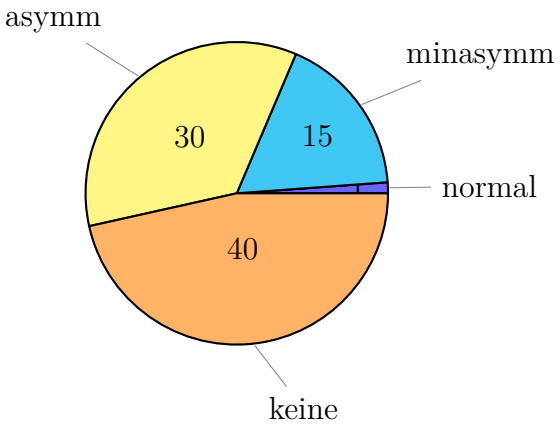
(a) Klassenverteilung für Modul Symmetrie



(b) Klassenverteilung für Modul Lidschluss



(c) Klassenverteilung für Modul Stirn



(d) Klassenverteilung für Modul Mund

Abbildung 10: Veranschaulichung der ungleichen Klassenverteilung jedes Modules (a-d) anhand des gegebenen Datensatzes. Bei einer Gleichverteilung müssen alle Teile gleich groß sein.

4.3 Vorgehensweise

In diesem Kapitel werden drei Vorgehensweisen für die Zusammensetzung der neun verschiedenen Bilder jede*r Patient*in, die während der Trainings- und Evaluierungsphase zum Einsatz kommt, erläutert. Zur Vereinfachung wird dies nur für die Modulform dargestellt. Analog gilt, dass diese Anwendungsformen auch für die direkte Gradermittlung (siehe Kapitel 4.2.2) genutzt wird. Für die direkte Ermittlung des Grades nach House-Brakckmann fällt der erste Schleifendurchgang für die einzelnen Module Symmetrie, Lidschluss, Stirn und Mund in den nachfolgenden Sequenzdiagrammen weg, da nur ein Neuronales Netz benötigt wird, das direkt den Grad feststellt.

4.3.1 Sequenzielles Verfahren

Die Anordnung der Bilder während der Trainings- und Evaluierungsschritte ist hierbei sequenziell. Das bedeutet im Trainingsprozess, dass jedes zugeschnittene Bild der neun Posen jede*r Patient*in für ein Modul hintereinander das Neuronale Netz durchlaufen,

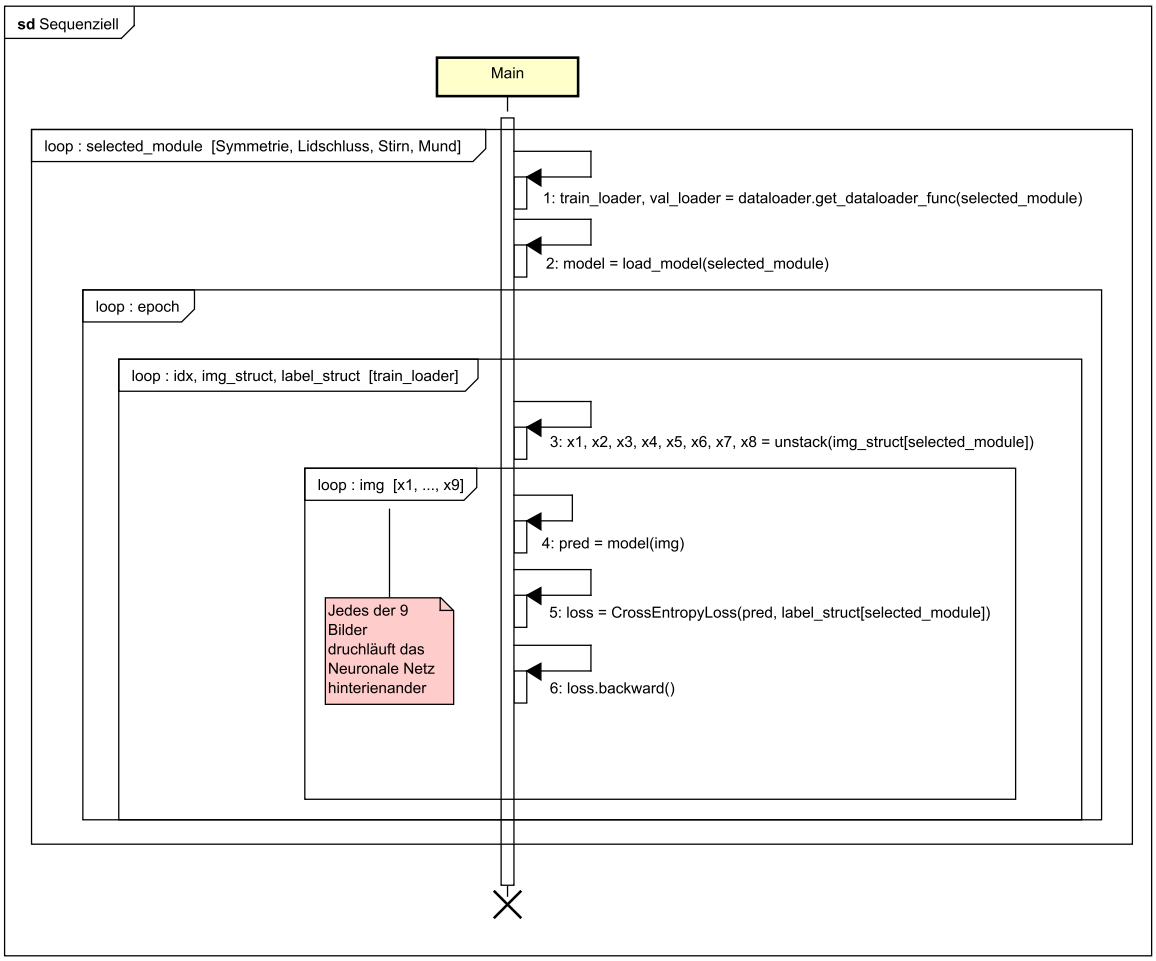


Abbildung 11: Sequenzdiagramm des Ablaufprozesses bei der sequenziellen Bearbeitung der neun Bilder. Diese durchlaufen hintereinander dasselbe Neuronale Netz.

bevor ein neuer Zyklus (Epoch) beginnt. Im dargestellten Sequenzdiagramm (siehe Abb. 11) ist zu sehen, dass nach der Wahrscheinlichkeitsermittlung für die Klassifikation des einzelnen Modules die Rückwärtsrechnung anhand des Losses das Neuronale Netz optimiert wird. Eines mögliches Problem liegt darin, dass z. B. der Trainingsfortschritt, der durch das erste Bild erzielt wurde, direkt vom nächsten wieder zerstört werden kann. Vorstellbar ist, dass die Posen, die in jedem Bild unterschiedlich dargestellt werden, eine komplementäre Wirkung auf das Modell erzielen.

Währenddessen wird im Evaluierungsprozess die beste Wahrscheinlichkeit der neun Durchgänge als das der wahren Klassifikation angesehen. So werden zu starke negative Auswirkungen von schlechteren Klassifikationswahrscheinlichkeiten vermieden.

Falls von den 9 Bildposen der/die Patient*in einzelne Posen fehlen und so während des Evaluierungs- und Trainingsprozesses nicht vorhanden sind, werden diese nicht beachtet. Der Schleifendurchlauf wird in einem solchen Fall abgebrochen, sodass die nicht vorhandene Pose keine Auswirkungen auf die Neuronalen Netze der Module haben.

4.3.2 Early Fusion

Eine andere Möglichkeit zur Eingabe in das Modell ist die Konkatenation der neun Bilder vorab. Die Hintereinanderreihung der drei Farbschichten RGB von den vorab zugeschnittenen Bildmaterialen (x_1 bis x_9) für ein spezifisches Modul erzeugt so einen Tensor mit der Größe $[27, a, b]$ mit a, b als Pixelgröße (siehe Abb. 12). Die Architektur der Neuronalen Netze ist dabei jedoch identisch zu dem sequenziellen Vorgehen. Danach werden die Tensoren in das Modell zur Klassifikation der Klassen des jeweiligen gerade bearbeiteten Modules hineingegeben. Ziel ist es, das Neuronale Netz selbst entscheiden zu lassen, welche Regionen zur Grad-Bestimmung nach House-Brackmann relevant sind.

Die nicht vorhandenen Bildcodierungen werden in dem Fall schwarz (Tensor mit nur den Werten 0) dargestellt bzw. generiert. So soll verhindert werden, dass sich die Parameter an den Kanten der Knoten in den Neuronalen Netzen verändern und der Fortschritt gestört wird.

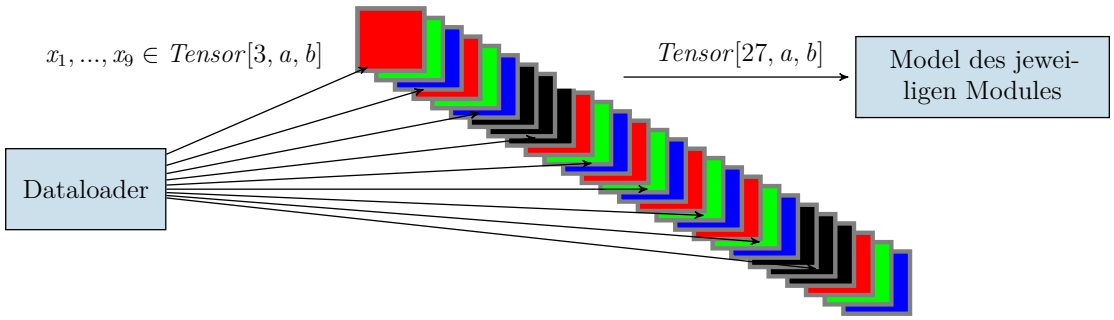


Abbildung 12: Darstellung der Konkatenation der neun zugeschnittenen Bilder eines Modules in Schichtform. Die schwarzen Schichten sind die aus den Bildern entstandenen Tensoren, gefüllt nur mit 0 Werten. Ausgabetensor nach der Operation wird dann in das Neuronale Netz für das jeweilige Modul gegeben.

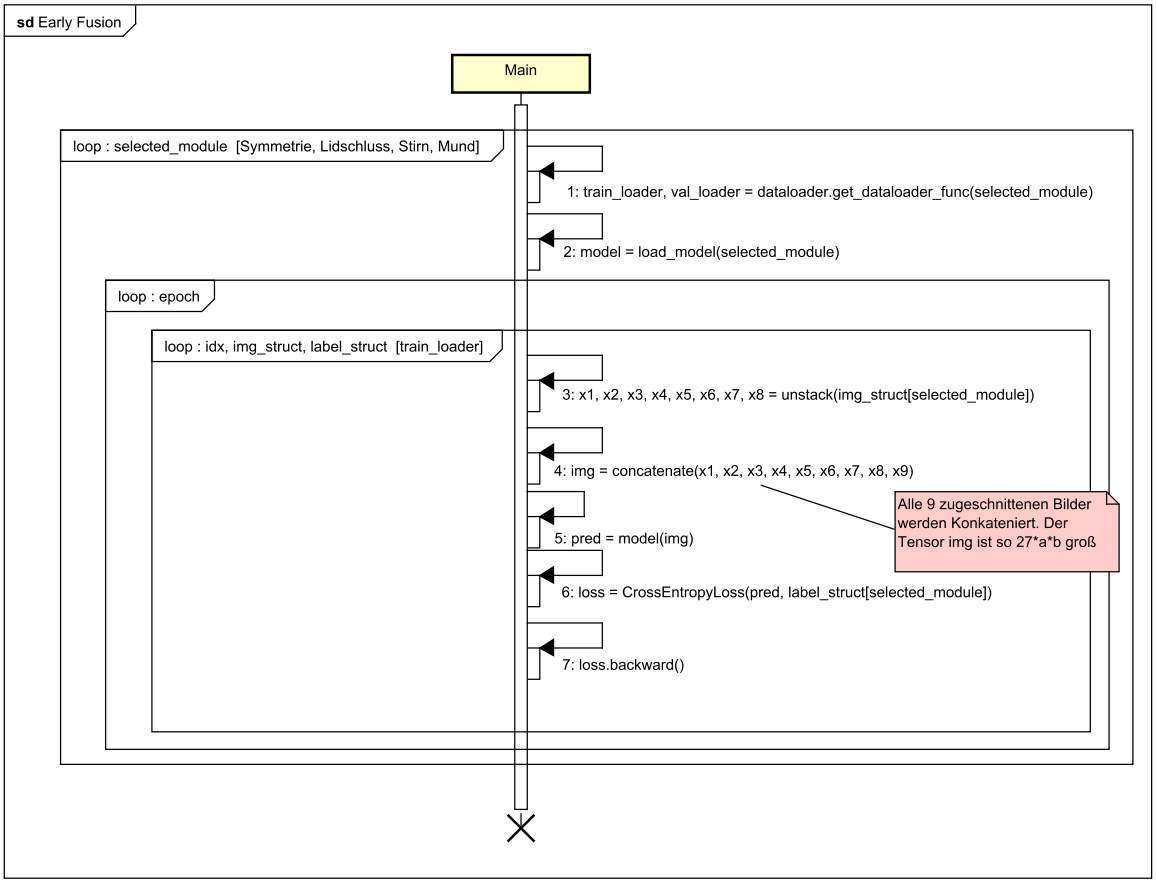


Abbildung 13: Sequenzdiagram des Ablaufprozesses bei Benutzung von Early Fusion. Dabei werden die neun zugeschnittenen Bilder für jedes Modul vorab konkateniert.

4.3.3 Late Fusion

Late Fusion ist das exakte Gegenteil von Early Fusion. Jedes Bild enthält ein eigenes Neuronales Netz für jedes Modul. Somit werden insgesamt bei vier Modulen und neun Bildcodierungen 36 separate Netze benötigt. All diese Netze müssen getrennt voneinander trainiert werden (siehe Abb. 15). Bei dem Evaluierungsprozess werden zunächst alle Modelle, die demselben Modul zugeordnet sind, fusioniert. Es gilt, dass die beste Klasse über alle neun Netze (siehe Abb. 14) diejenige ist, die für die spätere Zusammenfügung zu dem House-Brackmann Grad genommen wird, da angenommen werden kann, dass diese am wahrscheinlichsten die Realität abbildet. Sinn dahinter ist, die Posen als eine Gruppe zu interpretieren, welche zusammengehört. Die Modelle sollen also spezifisch auf die eingegebene Pose in die Sub-Netze reagieren und den Loss minimieren.

Dadurch, dass nicht alle Bildcodierungen für alle Patient*innen vorhanden sind, ist es notwendig, diese aus dem Trainingsprozess zu entfernen. Dazu kann, wenn eines der neun Bilder nicht vorhanden ist, der Schleifendurchgang übersprungen werden.

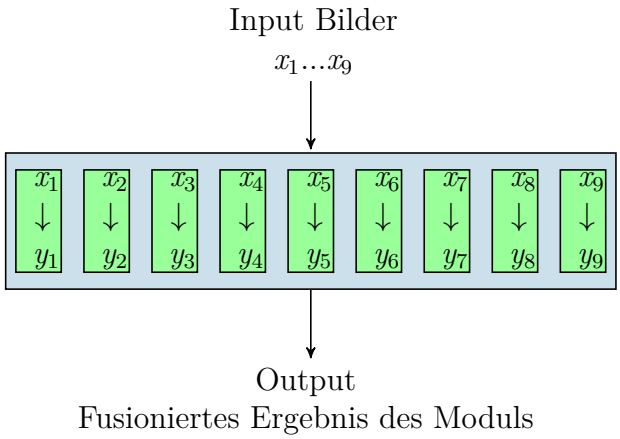


Abbildung 14: Darstellung eines Modules mit den neun Neuronalen Netzen. Der hellblaue Rahmen repräsentiert dabei ein Hauptmodul (Symmetrie, Lidschluss, Stirn, Mund), die grünen Rechtecke sind dabei die Sub-Netze für die Bilder x_1 - x_9. Das Ergebnis wird im Anschluss vor der Rückgabe fusioniert.

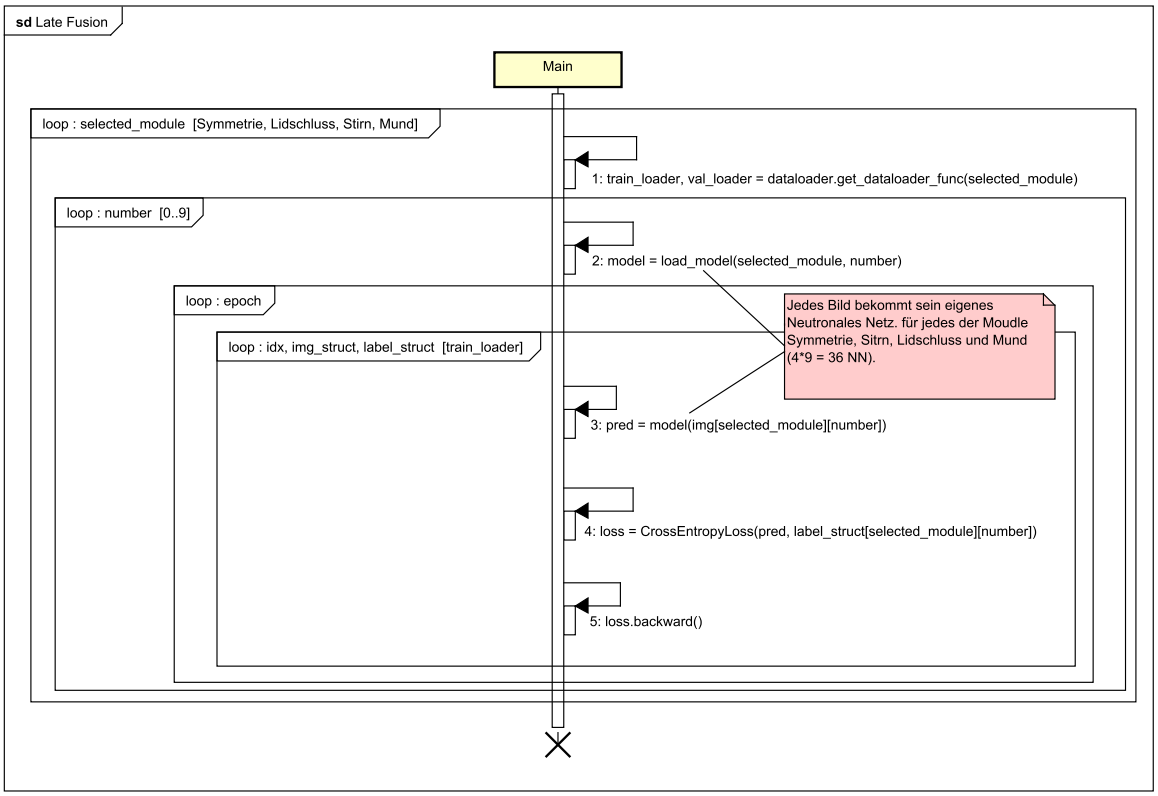


Abbildung 15: Sequenzdiagram des Ablaufprozesses bei Benutzung von Late Fusion. Hier existiert für jedes Bild ein separates Modell. Für jeden Modus sind so neun Modelle im Einsatz, woraus insgesamt 36 unabhängige Neuronale Netze entstehen.

4.4 Fusionierung der Module

Für die Fusionierung der einzelnen Module des zu ermittelnden Grades nach House-Brackmann werden zwei verschiedene Vorgehensweisen vorgestellt. Zunächst soll mithilfe eines Automaten anhand der besten Klassen jedes Modules der Grad festgestellt werden (Kapitel 4.4.1). Die zweite (direktere) Möglichkeit ist das Ausrechnen der Zeilensumme mit den direkt ausgegebenen Wahrscheinlichkeiten der einzelnen Klassen von den Neuronalen Netzen der Module (Kapitel 4.4.2).

4.4.1 Automat

Der Grad der Fazialisparese nach House-Brackmann kann durch einen Automaten bestimmt werden. Zuerst werden die Ausgabewahrscheinlichkeiten der einzelnen Module in die wahrscheinlichste Klasse umgewandelt. Die Einzelergebnisse der Module dienen als Eingabe in den Automaten. Das Wort, dass in den Automaten hineingegeben werden kann, setzt sich aus den Bestandteilen y_s, y_f, y_m, y_e zusammen. Die Funktionsnahmen sind hierbei die Wörter und die Ergebnisse die Gewichte an den Kanten. Der Endzustand des Automaten ist der zu ermittelnde Grad nach House-Brackmann.

Fix ist das Eingabewort $«y_s y_e y_f y_m»$, das für exakt diesen nichtdeterministischen Automaten (siehe Abb. 16) eingegeben werden muss, damit die Reihenfolge gewährleistet ist. Durch den Nichtdeterminismus wird entschieden, dass der nächste Buchstabe an der Reihe ist, wenn keine Änderung des Zustandes mehr möglich ist. So müssen auch nicht alle Transitionsfunktionen zwischen den Kanten und Knoten definiert werden. Leermengen in der Transitionstabelle (siehe Tabelle 4) sind die nicht definierten Kanten. Der erste Eingabebuchstabe wird aus der Eingabe genommen ($«y_s y_e y_f y_m» \rightarrow «y_e y_f y_m»$). Dieser Prozess wird so lange wiederholt, bis das Eingabewort vollständig abgearbeitet ist. Zu beachten sind dabei die Übergangsfunktionen zwischen zwei Zuständen. Jeder Buchstabe im Eingabewort hat eine Gewichtung (Ergebnis der Funktion), die entscheidet, ob der Pfad passierbar ist (siehe Abb. 16). Der Startzustand in den Automaten ist q_I, welche den niedrigsten anzunehmenden Grad nach der House-Brackmann Tabelle darstellen soll. Die Endzustände q_I - q_{VI} sind alle möglichen Grade der Skala.

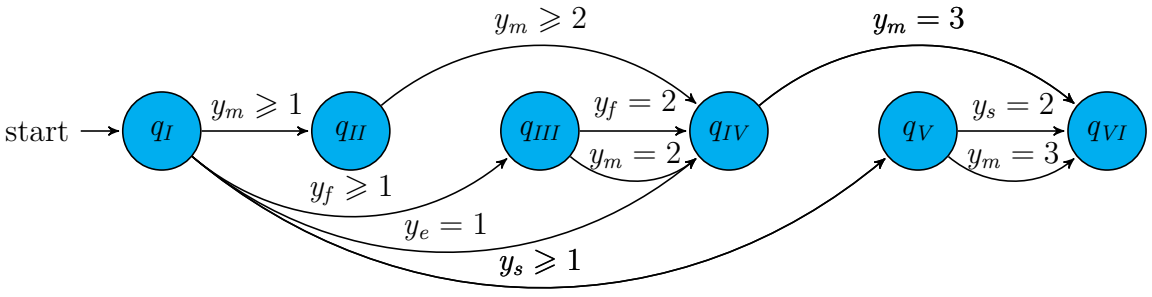


Abbildung 16: Nichtdeterministischer endlicher Automat zur Ermittlung der Schweregrades nach House-Brackmann mit dargestellten Gewichten der Module.

Q \ Σ	y_s	y_e	y_m	y_f
q_I	q_V, wenn $y_s \geqslant 1$	q_{IV}, wenn $y_e = 1$	q_{II}, wenn $y_m \geqslant 1$	q_{III}, wenn $y_f \geqslant 1$
q_{II}	$\varnothing$	$\varnothing$	q_{IV}, wenn $y_m \geqslant 2$	$\varnothing$
q_{III}	$\varnothing$	$\varnothing$	q_{IV}, wenn $y_m = 2$	q_{IV}, wenn $y_f = 2$
q_{IV}	$\varnothing$	$\varnothing$	q_{VI}, wenn $y_m = 3$	$\varnothing$
q_V	q_{VI}, wenn $y_s = 2$	$\varnothing$	q_{VI}, wenn $y_m = 3$	$\varnothing$
q_{VI}	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

Tabelle 4: Übergangsfunktion δ des Automaten für die Bestimmung des Grades.

Der NEA für die Bestimmung des Grades nach der Hosue-Brackmann Skala besteht aus dem Fünftupel $N = (Q, \Sigma, \delta, E, F)$:

- $Q = \{q_I, q_{II}, q_{III}, q_{IV}, q_V, q_{VI}\}$ ist eine endliche Zustandsmenge, wobei der Index repräsentativ der Grad ist, der ermittelt werden soll.
- $\Sigma = \{y_s, y_f, y_m, y_e\}$ ist ein endliches Alphabet (speziell begrenzt auf die Module)
- $\delta : Q \times \Sigma \to P(Q)$ Nicht-Deterministische Übergangsfunktion (siehe Tabelle 4)
- $E \subseteq Q = \{q_I\}$ ist der Startzustand
- $F \subseteq Q = Q$ Menge aller akzeptierten Endzustände

4.4.2 Gradermittlung durch Zeilensumme

Eine weitere Möglichkeit ist die Bildung von der Zeilensumme aus den einzelnen vorher definierten Modulen. Dazu werden die Klassenwahrscheinlichkeiten, die das Ergebnis der ausgeführten Prediction der Neuronalen Netze sind, an der Stelle ihres Modules (Spalte der Tabelle) und der Klasse in die abgewandelten House-Brackmann Tabelle eingetragen. Nachdem das für alle Module geschehen ist, werden die Zeilen addiert (siehe Tabelle 5).

Das höchste Ergebnis der sechs Zeilensummen ist der für den/die Patient*in angenommene Grad. Die Zeile ohne die Einzelwahrscheinlichkeit wird als Ausgangsergebnis zurückgeliefert.

Grad	Symmetrie		Stirn		Lidschluss		Mund		Σ
	Klasse	y_s	Klasse	y_f	Klasse	y_l	Klasse	y_m	
I	normal	0.20	normal	0.02	vollständig	0.30	normal	0.19	0.71
II	normal	0.20	normal	0.02	vollständig	0.30	minasymm	0.18	0.70
III	normal	0.20	minasymm	0.04	vollständig	0.30	minasymm	0.18	0.72
IV	normal	0.40	keine	0.05	unvollständig	0.46	asymm	0.16	**1.07**
V	asymm	0.16	keine	0.05	unvollständig	0.46	asymm	0.16	0.83
VI	keine	0.13	keine	0.05	unvollständig	0.46	keine	0.14	0.78

Tabelle 5: Darstellung der Einzelwahrscheinlichkeit aller Module und ihrer Klassen für die Zeilensummenbildung. Für dieses Beispiel ist Grad IV der als Resultat ausgegebene Grad nach House-Brackmann.

4.5 Caching

Problematisch ist bei größer werdenden Datensätzen, dass die Laufzeit schnell sehr zunehmen kann. Dazu muss der Server, der die Trainingsphase der Neuronalen Netze betreut, für alle Epochs bei allen Bildern die Berechnung der Punkte ausführen, die Bilder in die Regions of Interest zerlegen und die Augmentierung durchführen. Außerdem muss für die ganzen ResNet18 Netze der Module, die Vorwärtsrechnung ausgeführt und Rückwärtsoptimierung anhand des Losses optimiert werden. Die Schritte Punkteberechnung und Einteilung sind für alle Epochs identisch. Die Idee zur Beschleunigung der Trainingsschritte ist, diese fertigen zerschnittenen Bildfragmente in einem Cache vorab abzuspeichern, um diese Schritte einzusparen.

Zwei mögliche Lösungsansätze sind Least Recently Used Cache (LRU-Cache) (Kapitel 4.5.1) und eine stationäre externe Datenbank (Kapitel 4.5.2), der als temporärer Cache während der Trainingsphase agiert.

4.5.1 Least Recently Used Cache

LRU-Cache bedeutet, der am längsten nicht zugegriffenen Eintrag wird aus dem Cache entfernt. Vergleichen lässt sich das mit einer Warteschlange mit begrenztem Platzinhalt. Aufgerufene Elemente reihen sich am Ende der Warteschlange ein (im dargestellten Array die linke Seite). Alle nachfolgenden wandern um eine Position nach vorne. Der Vorderste in der Warteschlange wird entfernt (siehe Abb. 17), sobald sich ein noch nicht bekanntes Element in die Warteschlange einreiht. Dieses neue Element muss zur Laufzeit initial berechnet werden. Danach ist es so lange verfügbar, bis es wieder herausgeschoben wurde.

Einträge im Cache besitzen einen Index. Über diesen wird auf die dahinterliegenden Elemente, das Bildmaterial der/die einzelnen Patient*innen, zugegriffen. Sei l die Länge des Caches und Σ Indizes (z.B. A-Z), so können insgesamt l Elemente im Cache für das schnelle Aufrufen vorgehalten werden. Alle nicht vorgehaltenen Einträge müssen berechnet werden. Die im Cache liegenden Elemente besitzen eine kürzere Aufrufdauer als diejenigen, die nicht im Cache sind. Die maximale Ausführungszeit (eng. Worst Case Execution Time, WCET), also die Zeit, welche benötigt wird, um ein Element aufzurufen, beträgt für:

- im Cache liegend: $WCET_{ges} = WCET_{cache}$
- nicht im Cache liegend: $WCET_{ges} = WCET_{cache} + WCET_{berechnung}$

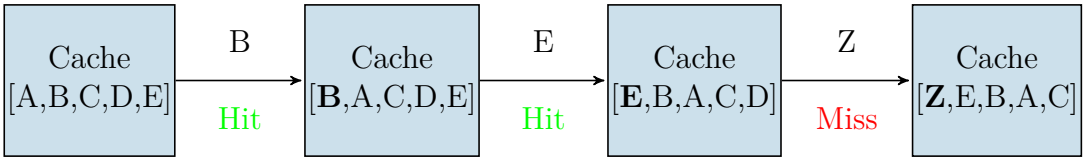


Abbildung 17: Hit sind Treffer. Diese Elemente sind bereits im Cache vorhanden. Miss hingegen sind dem Cache nicht bekannt, diese werden berechnet und reihen sich ein. Beispiel des LRU-Cache mit $l = 5$ und dem $\Sigma = A - Z$. Wenn Z sich einreiht, fliegt D raus, da kein Platz mehr vorhanden ist.

Dabei ist $WCET_{cache}$ die Laufzeit, die für das Nachschauen der Elemente im Cache benötigt wird, und $WCET_{berechnung}$ für die initiale Berechnung des abzuspeichernden Elementes beträgt. Solange l nicht wesentlich kleiner als die Anzahl von allen Elementen im Dataloader ist, können größtenteils Einträge wiederverwertet werden, wenn ein neuer Epoch in der Trainingsphase der Neuronalen Netze beginnt. Die Berechnung der recycelten Elemente kann demzufolge eingespart werden, woraus eine kürzere Gesamtlaufzeit zu erwarten ist.

4.5.2 Externe Datenbank

Eine weitere Möglichkeit besteht darin, Elemente in eine relationale Datenbank zu speichern (siehe Tabelle 6). Der Aufbau ist dabei relativ identisch zu dem LRU-Cache. Über einen Index wird auf das benötigte Element zugegriffen. Jedoch sind **alle Elemente** in der Datenbank gespeichert. Die WCET beträgt schlussendlich immer die Zeit, welche zum Zugriff auf die Datenbank benötigt wird. Zusätzlich kommt auf die globale Laufzeit einmal die Rechenzeit hinzu, welche benötigt wird, um alle Elemente zu berechnen und in die Datenbank einzutragen. Solange nicht auf denselben Index geschrieben wird, ist es ratsam, die Einträge parallel in die Datenbank zu schreiben und zu berechnen. Das hat den Vorteil, dass das komplette Potential des Servers/Computers ausgenutzt werden kann. Die Indexe repräsentieren dabei eine Nummer, die jedem einzelnen Patient*in zugeordnet ist. Indexe müssen als Primärschlüssel eineindeutig sein, damit keine Konflikte beim Schreiben und Lesen auftreten.

Indizes (Primärschlüssel)	**Elemente**
A	Bild 1
B	Bild 2
C	Bild 3
D	Bild 4
E	Bild 5
...	...

Tabelle 6: Beispieltabelle in einer relationalen Datenbank. Jeder Index besitzt genau ein Element, das ihm zugeordnet ist. Bei einem Zugriff auf den Index wird das Element als Rückgabewert ausgegeben.

Vorteil dieser Methode ist es, die kurze Lesezeit von der Datenbank ($WCET_{read}$) auszunutzen. Diese ist wesentlich kleiner als die Berechnung der Landmarks und das Einteilen der Regionen zur Laufzeit ($WCET_{berechnung}$). In der Trainingsphase werden z Epochs durchlaufen. Dadurch werden statt $WCET_{berechnung} * z$ nur $WCET_{read} * z$ Zeiteinheiten benötigt, wobei gilt $WCET_{berechnung} > WCET_{read}$. Dadurch kann die Trainingsphase schneller beendet oder auch ermöglicht werden, in der selben Laufzeit, mehr Epochs unterzubringen.

5 Experimente und Ergebnisse

5.1 Evaluation der Gesichtserkennung

Für die Genauigkeit der Position der Landmarks, im Verhältnis zu der Gesamtheit aller verfügbaren Patient*innen, wird für die neun Bilder jedes einzelnen Patienten, zu allererst die Ausrechnung dieser vollzogen. Festgestellt wurde, dass die Seitenverhältnisse und Größe der einzelnen Bilder des Datensatzes für das verwendete Framework „Face-Alignment“ von Adrian Bulat und Georgios Tzimiropoulos eine zu hohe Auflösung aufweisen und so die Landmarks nicht korrekt positioniert sind (siehe Tabelle 7). Durch Reverse Engineering des Sourcecodes wurde festgestellt, dass die verwendeten Bildmaterialen zum Trainieren des Neuronalen Netzes des Frameworks indirekt eine Standartauflösung von maximal 1920x1080px (HD) verwendet wurden [13].

Die Lösung für das Problem ist eine Verkleinerung der Größenverhältnisse in Abhängigkeit der tatsächlichen Bildgröße (a, b). Dabei ist a die Pixelanzahl in horizontaler und b in vertikaler Richtung. Die Größe der Bilder wird dazu in den Bereich für die optimale Ausführung zur Generierung der Landmarks verkleinert. Der Faktor F_{ab} für die Änderung der Bildgröße lässt sich wie folgt berechnen:

$$F(a,b) = \begin{cases} \frac{\max(a,b)}{10^3} + 1, & \text{wenn } \max(a,b) \mod 2 \\ \frac{\max(a,b)}{10^3}, & \text{sonst} \end{cases} \tag{9}$$

Nachdem die Landmarks vom System ausgerechnet wurden, werden diese auf das Originalbild mit der vollen Größe angewendet. Dazu werden alle Punkte mit F_{ab} multipliziert. Für die spätere Anwendung in den Neuronalen-Netzen und der Ausschneidung der Regionen werden so die hochauflösenden Bilder verwendet. Damit auch die Patientenbilder, deren Landmarks teilweise von der Ideallinie abweichen, verwendet werden können, muss beim Ausscheiden ein Offset hinzugerechnet werden.

		Platzierung der Landmarks		
		korrekt	teilweise	falsch
vor Anpassung	#	0	12	639
der Bildgröße	in %	0	1.84	**98.16**
nach Anpassung	#	572	78	1
der Bildgröße	in %	**87.87**	11.98	0.15

Tabelle 7: Plazierung der Landmarks vor und nach der Anpassung der Bildgröße durch den Faktor F_{ab} bezogen auf die 86 Patient*innen des Datensatzes und deren vorhandenen Bilder.

Des Weiteren kann die Rotation der Bilder mithilfe der Landmarks herausgefunden und falls nötig korrigiert werden. Dazu werden die absoluten Positionen von zwei Markern miteinander verglichen. Experimental lässt sich der Rotationswinkel R gegen den Uhrzeigersinn durch einfaches Ausprobieren mit den Punkten 0 (linkes Ohr) und 8 (Kinn) so definieren:

$$R[P(0), P(8)] = \begin{cases} 270°, & \text{wenn } P_a(0) < P_a(8) \wedge P_b(0) > P_b(8) \\ 180°, & \text{wenn } P_a(0) > P_a(8) \wedge P_b(0) > P_b(8) \\ 90°, & \text{wenn } P_a(0) > P_a(8) \wedge P_b(0) < P_b(8) \\ 0°, & \text{sonst} \end{cases} \tag{10}$$

5.2 Hyperparameter

Im folgenden Kapitel sollen kurz die grundlegenden Hyperparameter für die durchgeführten Experimente erläutert werden. Dazu zählen die Teilung des Datensatzes, Augmentierung der Bilder, die verwendete Lernrate und die Berechnung des Losses. Diese sind im späteren Verlauf für alle Experimente identisch, damit diese vergleichbar in ihrem jeweiligen Verhalten sind. Auch zu den Hyperparametern zählen die Anzahl an Epochs, die jedes Neuronale Netz zum Optimieren und Ausführen bekommt, und die Anzahl der Batchsize, die es den Netzen ermöglicht, parallel zu rechnen. Die Epochs sind auf 150 Durchläufe und die Batchsize auf 16, für die parallele Berechnung innerhalb der Neuronalen Netze, eingestellt.

Teilung des Datensatzes in zwei dusjunkte Hälften Die Teilung ist ein zwingendes Mittel, damit die Neuronalen Netze nicht den Datensatz auswendig lernen und in der realen Anwendung die Detektion der Klassen, ohne dass die Zielklasse bekannt ist, die Eingabedaten falsche, unwahre Ergebnisse zurückliefern. Dazu werden Mengen (siehe Abb. 18) aus dem vorhandenen Datensatz D ausgeschnitten. Trainingsdatensatz T und Validierungsdatensatz V sind echte disjunkte Teilmengen (11). Somit besitzen sie keine Schnittmenge. Das bedeutet wiederum, dass der Trainings- und Validierungsdatensatz keine gleichen Patient*innen haben. Der Trainingsdatensatz sollte dabei den größeren Gesamtanteil ausmachen. 0.75 hat sich dabei als bester Teilungsfaktor bewiesen. Randomisiert werden so aus dem gegebenen 86 Patient*innen 75%, also aufgerundet 65 Patient*innen, zugeordnet. Die anderen 25% (21 Patient*innen) werden dem Validierungsdatensatz hinzugefügt. Problematisch an der Teilung des Datensatzes ist, durch die zu kleine Anzahl an Patient*innen und ihren zugeordneten Graden, das Ungleichgewicht der Verteilung dieser, die dafür sorgt, dass nicht immer alle Grade in beiden Datensätzen zur Verfügung stehen. Die Lösung dafür ist es, falls zu wenige Grade des House-Brackmann Skalas im Trainings- oder Validierungsdatensatzes auftauchen, diese nicht mehr zu berücksichtigen.

$$\begin{aligned} T &\subset D \\ V &\subset D \\ T \cap V &= \varnothing \end{aligned} \tag{11}$$

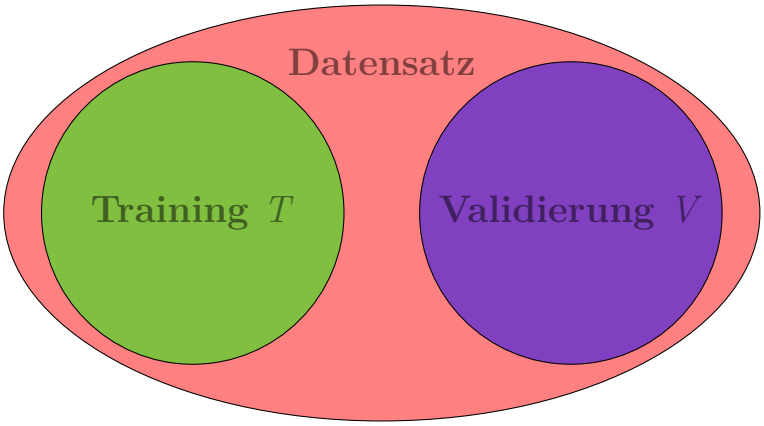


Abbildung 18: Disjunkte Mengen des Datensatzes Training T und Validierung V haben nicht die gleichen Patient*innen in ihren Mengen. Formal Mathematisch durch die Formel (11) definiert.

Augmentierung Um eine Vielfalt an verschiedenen Trainingsdaten zu erzielen, wird Augmentierung angewendet. Jeden Epoch, wenn die Neuronalen Netze ihr Training vollziehen, werden eine Reihe von Transformationen auf die neun Bilder jede*rs Patient*in durchgeführt. Eine Farbraumverschiebung der Pixelschichten Rot, Grün, Blau wird angewendet. Auch werden der Kontrast und die Helligkeit der Bilder minimal verändert. So werden verschiedene Hautfarben dargestellt. Der Zweck ist, dem Neuronalen Netz beizubringen, dass die Farbe nur im Hintergrund für die Detektion des House-Brackmann Grades von Bedeutung ist. Des Weiteren werden die Bilder gekippt, verschoben und um die Vertikalachse (Nasenrücken) gespiegelt, damit die ausgeprägte Seite der Fazialisparese prinzipiell egal ist. Das geschieht mit einer vordefinierten Wahrscheinlichkeit von 50%. Zusätzlich werden Verschiebungen und Rotationen simuliert, dass die Ausschnitte der Bilder nicht immer an der optimalen Stelle erfolgen. Auch wird der Schärfefaktor anhand eines Gauß Filters leicht unscharf gemacht. Damit werden fokussierte und unfokussierte Bilder erstellt, die einen Ungenauigkeitsfaktor erzeugen, der einen Fehler von Menschenhand nachstellen soll. Jedes der Module benötigt auch noch für ihre zugeschnittenen Ausschnitte aus den Bildern eine fixe Pixelgröße, anhand derer die Bilder verkleinert oder vergrößert werden (siehe Tabelle 8). Die neue Größe hat den Zweck, dass bei einer Batchsize, die Größe der parallelen Rechnungen der Neuronalen Netze ein einheitliches Format haben. Die Bilder werden auch noch zum Schluss normalisiert und in einen Tensor umgewandelt. Das drückt die Werte der einzelnen Pixelschichten von 0-255 in den Bereich zwischen -1 und 1. Dieser Schritt wird benötigt, damit die Faltungslayer im Neuronalen Netz besser funktionieren und performt. Die verschiedenen Transformationen werden nicht an den Validierungsdatensatz verwendet. Dieser bleibt unangetastet.

Modul	Symmetrie	Lidschluss	Stirn	Mund	Direkt
neue Pixelgröße der Bilder	640x640	420x500	640x420	640x300	640x640

Tabelle 8: Neue Pixelgröße für die einzelnen Module nach dem verkleinern oder vergrößern.

Ausrechnen des Losses Anhand der Gleichung (12) kann der Cross Entropy Loss zwischen der real zu ermittelnden Klasse und der ausgerechneten Klasse, auch bekannt als Prediction, ausgerechnet werden. Dazu werden sie als Tensor in die Formel eingesetzt. Dieser liefert so einen Wert, anhand dessen die Rückwärtsrechnung (eng. Backpropagation) ausgeführt wird und die Netze dahingehend optimiert werden, die richtige Klasse zu finden. Für die Modulform und der direkten Ermittlung des Grades ist die Größe C an die Anzahl der möglichen Klassen für das jeweilige Modul gekoppelt [14].

- C ist die Anzahl der Klassen, indiziert von $[0, ..., C-1]$
- $N = 16$ ist die Größe der Batchsize
- r sind die realen, vorgegebenen Klassen als Tensor mit C Einträgen
- p ist die Prediction aus den Neuronalen Netzen als Tensor mit C Einträgen

$$loss(p, r) = \{l_1, ..., l_N\}^T = \sum_{n=1}^{N} -\log \frac{\exp(p_{n,r_n})}{\sum_{c=1}^{C} \exp(p_{n,c})} \tag{12}$$

Lernrate Die Lernrate bildet einen der wichtigsten Hyperparameter. Diese Größe bestimmt die Strittveränderung an den Kanten und Knoten in den Neuronalen Netzen. Dieser Wert soll dabei helfen, dass die Netze schneller konvergieren. Es skaliert die Größe der Gewichtsaktualisierungen anhand des ausgerechneten Losses zwischen realer Klasse und der Prediction. Die Wahl der richtigen Größe der Lernnate kann schwierig sein. Ein zu kleiner Wert sorgt dafür, dass ein langer Trainingsprozess dazu führt oder im schlimmsten Fall das System stecken bleibt und keinen Fortschritt mehr erzielt. Zu große Werte hingegen sorgen dafür, dass der Trainingsprozess instabil wird. Für die angestrebten Experimente wird eine Exponentialfunktion, die von einer Cosinusfunktion überlagert wird, genutzt (siehe Abb. 19). Die so ausgewählte Lernnrate pro Epoch hat dadurch den besten Erfolg erzielt.

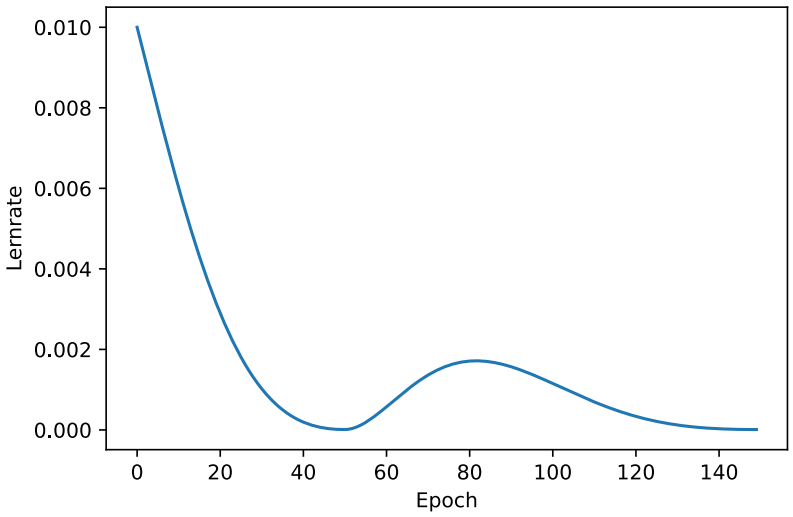
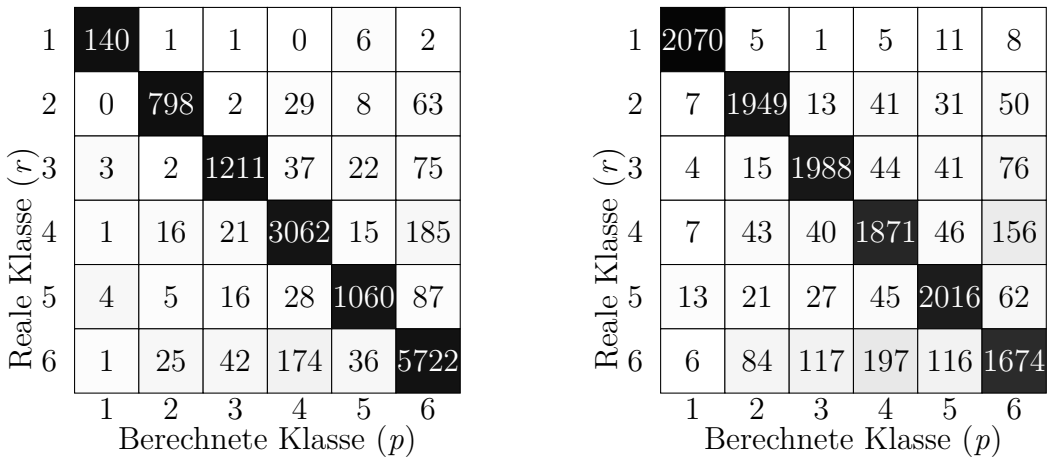


Abbildung 19: Lernrate in Ahbängigkeit des jeweiligen Epochs. Dabei werden eine Exponentialfunktion mit einer Cosinusfunktion miteinander verschmolzen. So werden periodische Peaks erzeugt. Diese erhöhen kurzfristig den Veränderungsfaktor der Parameter der Neuronalen Netze.

5.3 Nachweis der Funktionalität von Oversampling

In diesem kurzen Kapitel soll bewiesen werden, dass Oversampling die Ungleichheit des vorhandenen Datensatzes (Kapitel 4.1) nach der beschriebenen Methode (Kapitel 4.2.3) korrigieren kann. Es werden zwei Durchgänge des Datensatzes, einmal ohne und mit Oversampling, durchgeführt. Dazu wird nicht die Modulform Anwendung finden, sondern die direkte Ermittlung der House-Brackmann Grade und der vorab Konkatenation der neun Bilder mit Early Fusion. Anhand einer speziellen Wahrheitsmatrix, welche über die 150 Epochs hoch iteriert wird, kann so die korrekte Funktion von Oversampling festgestellt werden. Dort werden die Prediction der Neuronalen Netze (Spalte) und das Reale Ergebnis der Klasse (Zeile) jede*rs Patient*in in die richtige Position eingetragen.

Der Unterschied ist direkt erkennbar. Alle Klassen sind mit Oversampling durchschnittlich fast gleich oft vorhanden (siehe Abb. 20). Da die Zeilensummen der realen Klassen im Gegensatz ohne die Verwendung von Oversampling ungefähr die gleiche Größe der Summen haben. Auch wurde die Klasse 6, deren Anzahl im Datensatz den höchsten Anteil ausmacht, runtergeregelt und die Klasse 1 mit dem kleinsten Teil hingegen so erhöht. Das Tortendiagramm in Abbildung 5 im Kapitel 4.1 ist durch Oversampling verändert worden, dass alle Grade bzw. Klassen prozentual gleichverteilt vorkommen. Dies hat zur Folge, dass die Tortenstücke entweder verkleinert oder vergrößert werden, bis alle Teilstücke, abgesehen von einem Ungenauigkeistfaktor, gleich groß sind. Die Doppelbenutzung von Patient*innen ist auch experimentell durch die Ausgabe an der Kommandozeile bestätigt worden. Diese werden zum Ausgleichen der Klassen randomisiert aus dem Datensatz, spezifisch nur diejenigen mit dem richtigen Grad, gezogen. Damit ist die Annahme, dass Oversampling die Klassen ausgleichen kann, bestätigt. Dieses Konzept funktioniert analog mit der Modulform und mit den drei Vorgehensweisen Sequenziell, Early Fusion und Late Fusion.



(a) ohne Oversampling **(b)** mit Oversampling

Abbildung 20: Wahrscheinlichkeitsmatrizen (eng. Confusion Matrix) mit und ohne Oversampling über alle Epochs bezogen. Klar erkennbar ist der Unterschied durch die höheren Werte an den Diagonalen. Die Summe der Zeilen sind nach dem Oversampling ungefähr gleich groß. Klasse 1-6 sind die Grade von I-VI der House-Brackmann Skala.

5.4 Sequenziell

Die sequenzielle Fütterung der zerschnitten neun Bilder jede*r Patient*in in die vier Neuronalen Netzen zeigt keinen nachweisbaren Erfolg für die richtige Klassifikation. Sowohl mit als auch ohne Oversampling liegt bei der Validierung der F1-Wert bei ca. 0.3 konstant für alle Epochs (siehe Abb. 21 und 22). Oversampling bei der direkten Ermittlung des House-Brackmann Grades hat in dem Fall die Auswirkung, dass wesentlich schneller höhere F1-Werte erzielt wurden. Oversampling hat so den Vorteil gebracht, dass weniger Epochs ausreichen um denselben F1-Wert zu erreichen, ohne es zu verwenden. Bei beiden Graphen ist auch klar ersichtlich, dass rund um den 80ten Epoch höhere Sprünge zwischen den Epochs stattfinden, welches durch den Cosinusanteil innerhalb der Lernrate bewirkt wird. Deutlich ist auch, dass die Maximalwerte erst ab den ca. 120ten Epoch erzielt worden sind.

Tabellarisch festellbar ist auch, dass die Anwendung von Oversampling im Trainingsdatensatz, bei der Modulform und der direkten Detektion, für alle statistischen Merkmale ein höherer Wert erreicht wurde (siehe Tabelle 9). Dieser ist um 0,193 anngestiegen für die Modulform und 0,062 für die direkte Ermittlung. In beiden Tabellen ist auch klar sichtbar, dass die Modulform immer schlechter ist als die direkte Ermittlung des Grades nach House-Brackmann (siehe Tabelle 10).

	Oversampling	Sensitivität TPR	Spezivität TNR	Positiver Vorhersagewert PPV	Negativer Vorhersagewert NPV	F1 Wert
Modulform	nein	0.424	0.682	0.436	0.752	0.414
	ja	0.625	0.806	0.631	0.817	0.607
Direkt	nein	0.810	0.963	0.924	0.973	0.854
	ja	0.912	0.983	0.930	0.983	0.916

Tabelle 9: Statistisch besten Werte der Metirken des Trainingsdatensatzes mit und ohne Oversampling bei sequenzieller Anordnung. Der beste F1-Wert wurde dabei von der direkten Ermittlung des Grades unter Benutzung von Oversampling getroffen. Klar erkennbar ist der positive Effekt von Oversampling während des Trainings.

	Oversampling	Sensitivität TPR	Spezivität TNR	Positiver Vorhersagewert PPV	Negativer Vorhersagewert NPV	F1 Wert
Modulform	nein	0.393	0.678	0.447	0.759	0.355
	ja	0.368	0.653	0.422	0.767	0.330
Direkt	nein	0.843	0.970	0.949	0.980	0.884
	ja	0.870	0.977	0.977	0.986	0.914

Tabelle 10: Statistisch besten Werte der Metirken des Validierungsdatensatzes mit und ohne Oversampling bei sequenzieller Anordnung. Der beste F1-Wert wurde dabei von der direkten Ermittlung des Grades unter Benutzung von Oversampling getroffen.

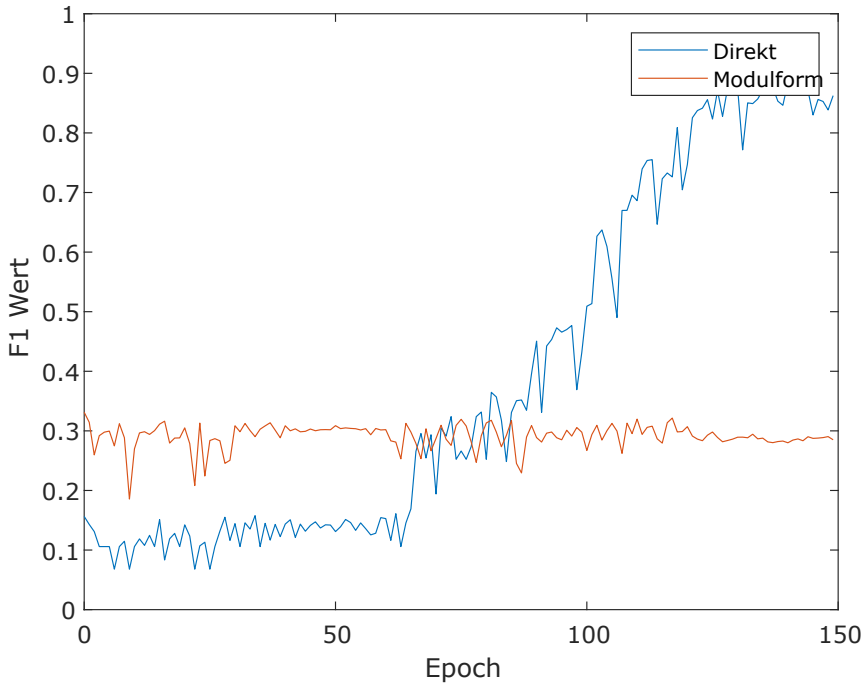


Abbildung 21: F1-Wert jedes Epochs des Validierungsdatensatzes mit der sequenziellen Verarbeitung der neun Bilder jede*r Patient*in ohne Anwendung von Oversampling.

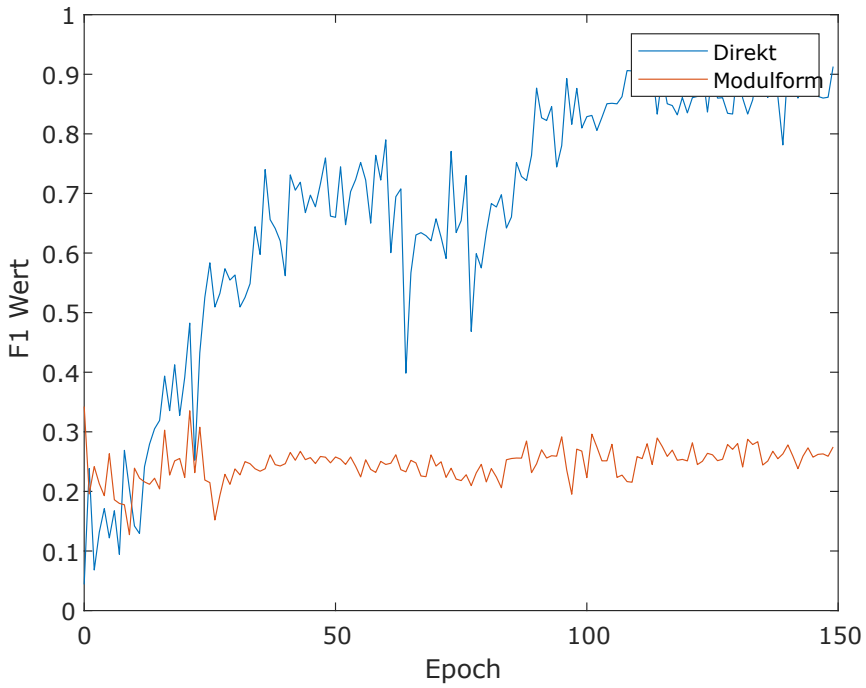


Abbildung 22: F1-Wert jedes Epochs des Validierungsdatensatzes mit der sequenziellen Verarbeitung der neun Bilder jede*r Patient*in unter Anwendung von Oversampling.

5.5 Early Fusion

Unter der Anwendung von Early Fusion, welches die neun Bilder jede* Patient*in vorab konkateniert und als eine Einheit in die jeweiligen Module oder für die direkte Ermittlung in die Neuronalen Netze hineingegeben wird, wirkt sich die Benutzung von Oversampling zum Klassenausgleich in keinerlei Art auf den maximalen Ausschlag der statistischen Merkmale aus (siehe Tabelle 11 und 12). Unterschiede zwischen dem Trainings-, dem Evaluierungsablauf und ihren zugeordneten Datensätze sind nicht merkbar festzustellen. Der F1-Wert von beiden, Validierungs- und Trainingsdatensatz, zeigen den Wert 1 an. Das bedeutet, dass ab einem diskreten Zeitpunkt (ohne ca. 35 Epochs, mit ca. 120 Epochs), immer eine korrekte Klassifizierung durchgeführt worden ist unter der Verwendung dieses Datensatzes. Dies gilt für sowohl mit als auch ohne Oversampling.

Das Nutzen von Oversampling hat hier einen großen Nachteil, was die benötigte Anzahl der Epochs angeht (siehe Abb. 23 und 24). Der Graph des F1-Wertes hinkt hinterher, wenn Oversampling genutzt wird. Bei Epoch 25 ist ohne Oversampling der Wert 0.95 erreicht worden. Mit Oversampling ist dieser erst bei ca. 0.85. Der gleiche Wert wie ohne Oversampling wurde erst mit dem ca. 120ten Epoch erreicht worden. Somit benötigt das Neuronale Netz, mit der Anwendung von Oversampling, 105 Epochs länger, um gleichwertig zu sein. Deutlich ist auch, dass kein merkbarer Unterschied zwischen der Modulform und der direkten Ermittlung existiert. Daraus kann geschlossen werden, dass die vier Module gleichwertig zu der direkten Ermittlung sind.

	Oversampling	Sensitivität TPR	Spezivität TNR	Positiver Vorhersagewert PPV	Negativer Vorhersagewert NPV	F1 Wert
Modulform	nein	0.975	0.968	0.931	0.997	0.977
	ja	0.978	0.993	0.980	0.994	0.978
Direkt	nein	1.000	1.000	1.000	1.000	1.000
	ja	1.000	1.000	1.000	1.000	1.000

Tabelle 11: Statistisch besten Werte der Metirken des Trainingsdatensatzes mit und ohne Oversampling mit Anwendung von Early Fusion. Die besten erzielten Werte wurden mit der direkten Ermittlung erreicht.

	Oversampling	Sensitivität TPR	Spezivität TNR	Positiver Vorhersagewert PPV	Negativer Vorhersagewert NPV	F1 Wert
Modulform	nein	0.979	0.998	0.989	0.987	0.980
	ja	0.974	0.986	0.966	0.984	0.967
Direkt	nein	1.000	1.000	1.000	1.000	1.000
	ja	1.000	1.000	1.000	1.000	1.000

Tabelle 12: Statistisch besten Werte der Metirken des Validierungsdatensatzes mit und ohne Oversampling mit Anwendung von Early Fusion. Die besten erzielten Werte wurden mit der direkten Ermittlung erreicht.

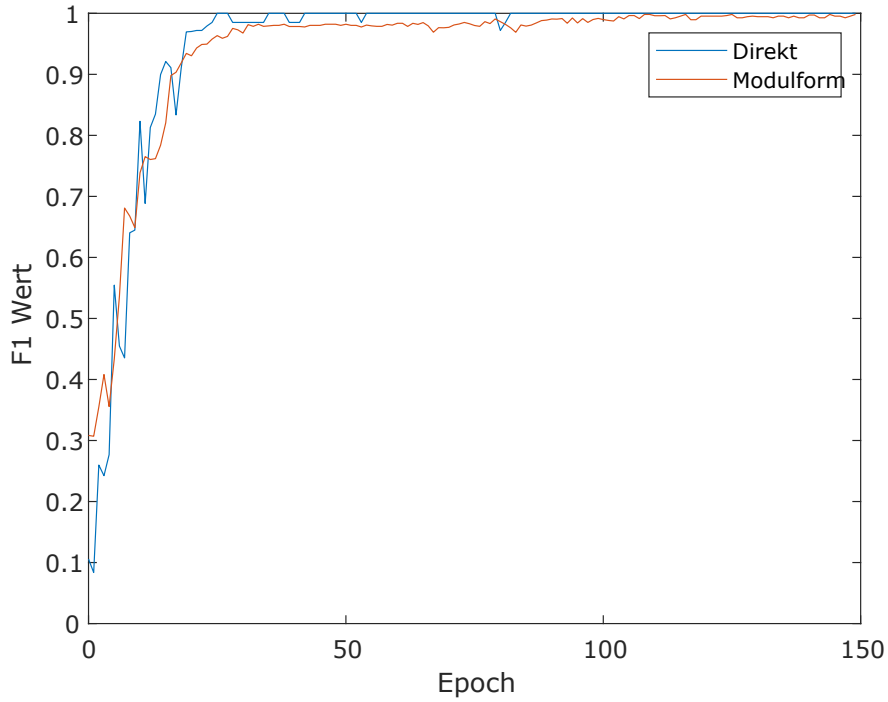


Abbildung 23: F1-Wert jedes Epochs des Validierungsdatensatzes mit Early Fusion der neun Bilder jede*r Patient*in ohne Anwendung von Oversampling.

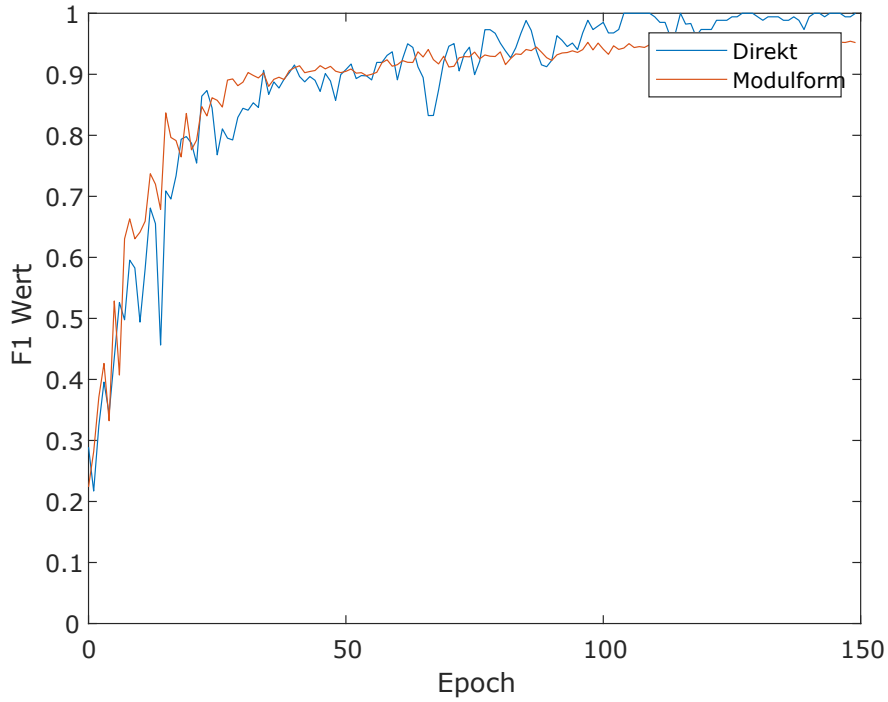


Abbildung 24: F1-Wert jedes Epochs des Validierungsdatensatzes mit Early Fusion der neun Bilder jede*r Patient*in unter Anwendung von Oversampling.

5.6 Late Fusion

Wie bereits dargestellt, ist Late Fusion das Nutzen von neun Neuronalen Netzen, die für jedes Bild ein Netz bereitstellen und das für jedes Modul. Es kann festgestellt werden, dass die F1-Kurve des Validierungsdatensatzes ohne Oversampling, für die Modulform und die direkte Detektion, ab den 40ten Epochs stagniert (siehe Abb. 25). Selbst mit der Erhöhung der Lernrate um den 80ten Epoch ist der Zuwachs so gering, dass es vernachlässigbar ist. Zwischen den ersten Epoch und den 25ten überholt der F1-Wert der direkten Ermittlung die Modulform. Mit Oversampling sind höhere Schwankungen zwischen den Epochs zu sehen. Jedoch ist der F1-Wert zwischen den Epoch 40 und 150 konstant wachsend (siehe Abb. 26).

Oversampling zum Klassenausgleich hat hier bei der direkten Ermittlung des Grades einen leicht höheren F1-Wert von 0.927 erzielt. Währenddessen sind bei der Modulform sowohl im Trainingsdatensatz als auch im Validierungsdatensatz die Metriken, bis auf kleine Ungenauigkeiten, gleich (siehe Tabelle 13 und 14).

	Oversampling	Sensitivität TPR	Spezivität TNR	Positiver Vorhersagewert PPV	Negativer Vorhersagewert NPV	F1 Wert
Modulform	nein	0.822	0.833	0.814	0.814	0.815
	ja	0.818	0.829	0.803	0.826	0.817
Direkt	nein	0.867	0.977	0.943	0.986	0.896
	ja	0.918	0.984	0.957	0.986	0.927

Tabelle 13: Statistisch besten Werte der Metirken des Trainingsdatensatzes mit und ohne Oversampling mit Anwendung von Late Fusion. Die besten erzielten Werte wurden mit der direkten Ermittlung erreicht.

	Oversampling	Sensitivität TPR	Spezivität TNR	Positiver Vorhersagewert PPV	Negativer Vorhersagewert NPV	F1 Wert
Modulform	nein	0.821	0.832	0.814	0.831	0.817
	ja	0.814	0.831	0.812	0.818	0.808
Direkt	nein	0.867	0.977	0.942	0.986	0.895
	ja	0.918	0.984	0.957	0.985	0.927

Tabelle 14: Statistisch besten Werte der Metirken des Validierungsdatensatzes mit und ohne Oversampling mit Anwendung von Late Fusion. Die besten erzielten Werte wurden mit der direkten Ermittlung erreicht.

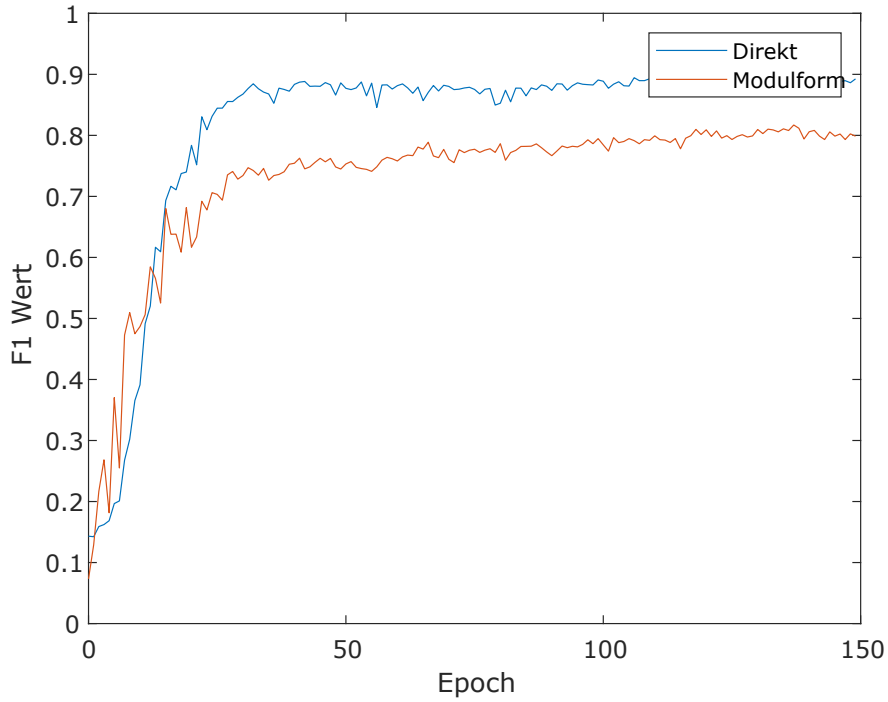


Abbildung 25: F1-Wert jedes Epochs des Validierungsdatensatzes mit Late Fusion der neun Bilder jede*r Patient*in ohne Anwendung von Oversampling.

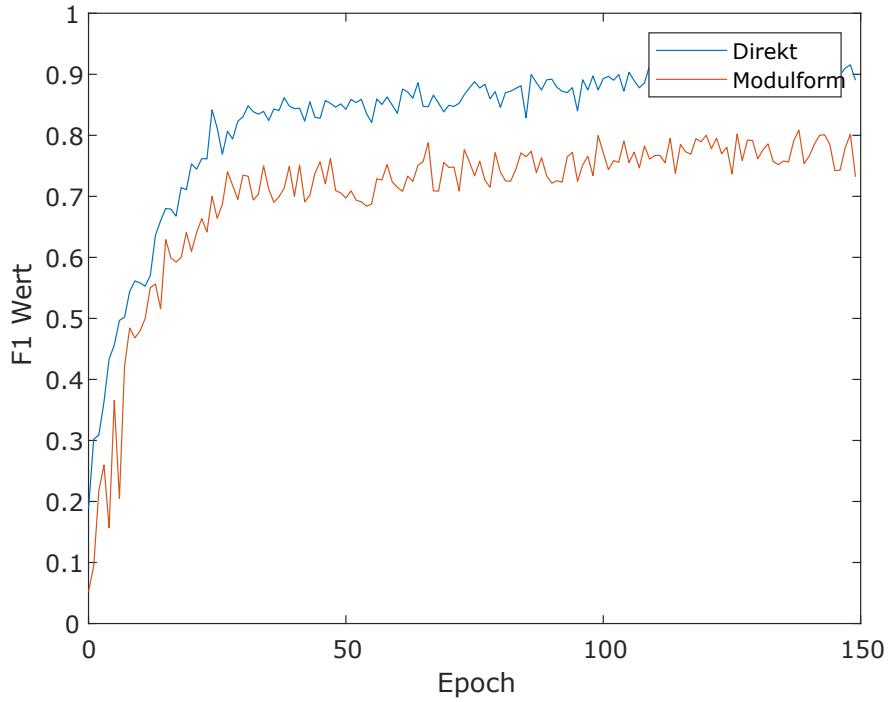


Abbildung 26: F1-Wert jedes Epochs des Validierungsdatensatzes mit Late Fusion der neun Bilder jede*r Patient*in unter Anwendung von Oversampling.

5.7 Laufzeitanalyse des Caches

Analysiert wurde die Laufzeiten des Systems bezogen auf fünf Epochs und eine reduzierte Anzahl von 53 Patient*innen in einem verkleinerten Datensatz. Die Größe des LRU-Cache wurde stückweise von 47 bis 55 jeweils um eins inkrementiert. Wenn die maximale Anzahl der im LRU-Cache platzhabenden Elemente größer ist als die Anzahl Patient*in im Datensatz, wird die Laufzeit stabil um einen Fixwert gehalten. Dieses Verhalten ist erklärbar dadurch, dass jedes Element des Datensatzes einen Platz im LRU-Cache bekommen kann. Damit wird keine Rechenzeit für die Berechnung verschwendet. Auch signifikant feststellbar ist der rapide Zuwachs an Zeiteinheiten, wenn die Größe des LRU-Cache abnimmt. Der Zeitzuwachs nicht linear und ähnelt einer Sigmoidfunktion. Die Hitrate (Trefferrate für im Cache liegende Elemente) nimmt mit verminderter Größe ab. Mehr Elemente müssen zur Laufzeit zuerst berechnet werden. Ab einem Punkt ist kein Zeitzuwachs mehr zu erwarten. Die Missrate ist so hoch, dass der LRU-Cache keine signifikante Auswirkung auf die Laufzeit hat (siehe Abb. 27).

Wenn zusätzlich zu dem LRU-Cache die Datenbank als externer Cache zugeschalten wird, ist die Laufzeit bis auf eine minimale Ungenauigkeit für genau dieselbe Anzahl an Epochs identisch. Diese kleinen Schwankungen in der Laufzeit beruhen auf der Auslastung der CPU sowie der Grafikkarte und können für die Betrachtung vernachlässigt werden.

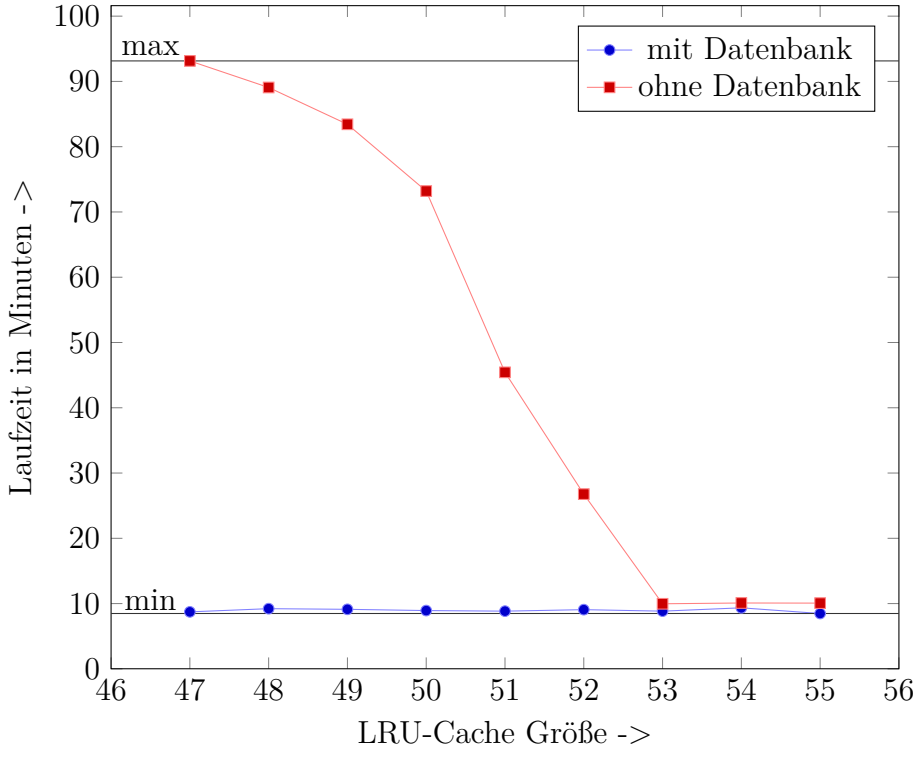


Abbildung 27: Laufzeiten des Systems (fünf Epochs, 53 Datensätze) mit und ohne Datenbank als Cache und variabler LRU-Cache Größe.

6 Diskussion und Ausblick

6.1 Vor- und Nachteile der direkten Ermittlung gegenüber dem Modulaufbau

Die Vorteile der Modulform liegen in der unabhängigen Funktionalität der Module. Diese besitzen ihre jeweils zugeschnittenen Bildmaterialien für die entsprechenden Module. Durch die unabhängige Funktion ist ebenso sichergestellt, dass, falls ein Modul für eine Person eine nicht korrekte Klassifikation ausführt, die anderen Module nicht von ihm beeinflusst werden. Ein weiterer Vorteil ist, dass durch den hochmodularen Aufbau einzelne Module ausgegliedert und als Expertensysteme separat in anderen Projekten ohne weiteres Zutun verwendet werden können. Einzige Voraussetzung hierbei ist, dass diese Systeme eine korrekte Klassifizierung ausführen.

Problematisch ist allerdings, dass, falls die Einteilung der Regions of Interest durch das verwendete Framework für die einzelnen Module nicht korrekt funktioniert und somit falsche Bereiche ausgewählt werden, keine Korrektur oder Fehleranalyse erfolgen kann. Dadurch kann es zu fehlerhaften Klassifikationen kommen, die so nicht feststellbar sind. Des Weiteren ist von Nachteil, dass die Ermittlung des Grades aus vier Modulen besteht, die im Nachhinein zu dem Grad nach House-Brackmann fusioniert werden. Die Schwachstelle dabei ist, dass verschiedene Wege zur Bildung des Grades führen. Dabei ist die Verwendung des Automaten oder die Nutzung der Zeilensumme der freien Entscheidung überlassen und somit wieder subjektiv in der Meinung des/der jeweiligen Anwender*ins. Das wiederum ist für Patient*innen von Nachteil, da der Ausgabegrad objektiv sein soll.

Die Anwendung der direkten Ermittlung des Grades hingegen hat den Vorteil, dass keine Fusionierung durch Automaten, Zeilensumme oder anderes nötig ist. Diese Methode basiert nur auf der Eingabe der neun Bilder der/die Patient*innen und den als Ausgabe rückzugebenden Grad. Auch ist so keine Modulbildung nötig, welches das Vierfache an Zeit an der obigen dargestellten Methode benötigt. So können auch besser echtzeitkritische Detektionen ausgeführt werden, welche für die Praxisanwendung vorteilhaft wären.

Nachteilhaft ist, dass anhand der direkten Ermittlung, falls ein Fehler in der Klassifikation stattfindet, einzig und allein durch das eine Neuronale Netz der direkten Ermittlung der Fehler verursacht werden kann. So wird umso wichtiger, dass das Training und die Validierung der Netze in einem Datensatz ausgeführt werden, welcher mehrere tausend Elemente beinhaltet. Die Sammlung einer solchen Anzahl an Patientendaten, welche ebenfalls noch Datenschutzkonform gesammelt werden müssen, ist eine weitere große Herausforderung.

6.2 Vor- und Nachteile der verschiedenen Vorgehensweisen

Die verschiedenen Betrachtungsweisen zur Fusion bzw. Abarbeitung der neun Bilder der/die Patient*innen haben ihre Vor- und Nachteile. Die sequenzielle Abarbeitung der Bilder ist zwar einfach in der Umsetzung. Jedoch ist denkbar, dass der Fortschritt der Neuronalen Netze zum Teil überschrieben und rückgängig gemacht wird. So kann zum Beispiel der Erfolg von Bild 1 durch die anderen acht Bilder zunichte gemacht werden. Sichtbar wird dies durch die sehr sprunghaften Veränderungen im F1-Graph und den erst verzögerten Anstieg bei der direkten Ermittlung.

Das Zerschneiden der Bilder in die Regions of Interest für die sequenzielle Fütterung der Neuronalen Netze der Module hat keine positiven Auswirkungen auf eine Verbesserung einer korrekteren Klassifizierung. Da kein Anstieg, weder mit noch ohne Oversampling zum Klassenausgleich, ersichtlich ist, scheint sich die Annahme, dass sich die neun Ausschnitte der Bilder für jedes Modul gegenseitig zu überschreiben, zu bestätigen.

Late Fusion mit der Nutzung von Oversampling hat den Nachteil, dass die Detektion sehr viel Zeit in Anspruch nimmt. Trotz dessen, dass die Modulform insgesamt 36 Neuronale Netze (pro Bild eins und das für die vier Module) nutzt, ist, im Gegensatz zu der sequenziellen Verarbeitung, ein relativ konstantes Wachstum zu verzeichnen. Für die direkte Ermittlung beim sequenziellen Verfahren sowie beider (Direkt und Modulform) für Late Fusion hätten mehr als die 150 Epochs benötigt, weil der F1-Wert noch kein Plateau bzw. die Kurve kein Abfallen aufzeigt. Mit mehr Epochs hätten die Neuronalen Netze die Chance, einen Zuwachs des F1-Wertes zu erzielen.

Das beste Ergebnis wurde sowohl für die Modulform als auch für die direkte Detektion mit der Nutzung von Early Fusion erzeugt. Anzunehmen ist, da der F1-Wert nahezu perfekt durch Oversampling bzw. eins ohne Anwendung von Oversampling ist, dass zu wenige Patient*innen im Datensatz enthalten sind. Der Verdacht liegt nahe, trotz der Trennung in einen Trainings- und Validierungsdatensatz, dass das Neuronale Netz die konkatenierten Bilder auswendig gelernt hat. Auch kann es sein, dass die Augmentierung der Bilder zu schwach eingestellt ist. Dagegen spricht, da es sich um neun verschiedene Posen handelt, die für jede*n Patient*in vorhanden sein sollten, vorab konkateniert und dem Netz überlassen worden ist, den richtigen Schluss zur richtigen Klasse des Grades ziehen soll.

Für alle Vorgehensweisen, sequenzielle Anordnung, Early und Late Fusion, für die vorgestellten Methoden Modulfom und direkten Ermittlung ist die Menge enthaltener Patient*innen im Datensatz zu gering, um ein stichhaltiges aussagekräftiges Ergebnis der Experimente zu erhalten. So wie die F1-Graphen der Experimente darstellen, kann eine Tendenz erkannt werden, dass generell die Möglichkeit besteht, mit den genannten Methoden durch das Nutzen von Neuronalen Netzen eine Ermittlung des Grades nach House- Brackmann durchführbar ist. Für ein aussagekräftiges Ergebnis zur Genauigkeit und Richtigkeit der festgestellten Klassen müssen die Experimente wiederholt werden. Die Neuausführung der Experimente sollte mit einen deutlichen größeren Datensatz erfolgen.

6.3 Zusammenfassung

Zusammenfassend wurden verschieden Experimente zu den Methoden, die House-Brackmann Tabelle als einzelne Module zu betrachten, und der direkten Ermittlung der Grade I - VI, welche die Einteilungsstufen der House-Brackmann Grade ist, durchgeführt. Dabei wurden Experimente zu den verschiedenen Vorgehensweisen zur Verarbeitung der neun Bilder vorgestellt bzw. durchgeführt. Die Vorgehensweisen sind die sequenzielle Anordnung, Early Fusion durch Konkatenation und Late Fusion, wobei jedes Bild ein eigenes Netz erhielt. Zum Ausgleich der nicht gleichverteilten Klassen der/die Patient*innen wurde Oversampling zum Klassenausgleich sowohl in der Modulform für die einzelnen Bestandteile der House-Brackmann Skala als auch in der direkten Ermittlung erfolgreich angewendet. Je nach Vorgehensweise hatte Oversampling eine andere Auswirkung auf die Schnelligkeit des Trainingserfolges, sichtbar durch den F1-Wert.

Der Vorteil der Modulform ist die unabhängige Nutzung in anderen medizinischen Bereichen. Dort kann, falls ein anderes Experiment oder eine andere Anwendung ein Modul z. B. Liedschluss als Kriterium benötigt, auch ohne die Ausführung der Detektion des Grades genutzt werden. So sind, durch den hoch modularen Aufbau alle Komponenten, Einteilung in die Regions of Interest, die Module sogar die Fusionierung des Grades einzeln oder auch im Gesamtpaket nutzbar. Nachteilhaft ist, dass die Modulform mehrere Neuronale Netze beinhaltet und deswegen wesentlich mehr Rechenzeit benötigt als die direkte Form der Ermittlung. Bewiesen wurde auch, dass es möglich ist, durch das aktive Benutzen von Caches mit den Implementierungsformen LRU-Cache und Datenbank, schneller die Trainingsphase der Neuronalen Netze durchlaufen zu lassen.

Projekte	**F1-Wert in %**
Insu et al. [15]	89.23
Hyun et al. [12]	87.11
Muhammad et al. [16]	92.91
Ting et al. End-to-End [10]	83.41
Ting et al. Kaskadierend [10]	95.97
Modulform Sequenziell (Unser)	35.50
Modulform Early Fusion (Unser)	98.00
Modulform Late Fusion (Unser)	92.70
Direkt Sequenziell (Unser)	91.40
Direkt Early Fusion (Unser)	100.00
Direkt Late Fusion (Unser)	92.70

Tabelle 15: Leistungsvergleiche der Graduierung der Fazialisparese in Bezug auf den F1-Wert. Einordnung zu anderen Experimenten und Projekten. Dazu sind jeweils die besten der Modulform und der Direkten Ermittlung genommen worden [10].

Zur Einordnung der Qualität und Funktionalität der durchgeführten Experimente wurden diese mit anderen Projekten verglichen (siehe Tabelle 15). Leider sind die Ergebnisse der Experimente nicht aussagekräftig genug, da zu wenig Patient*innen sowohl im Trainings- als auch im Validierungsdatensatz enthalten sind. Der Verdacht liegt nahe, dass die Neuronalen Netze bei der Anwendung von Early und Late Fusion sowie bei der sequenziellen Anordnung die Bilder auswendig gelernt oder die Augmentierung zu schwach eingestellt war. Der Vorteil, im Gegensatz zu den anderen Projekten, ist die Nutzung von neun verschiedenen Posen, wodurch theoretisch die Feststellung des Grades nach House-Brackmann leichter sein sollte.

6.4 Ausblick

Es gibt verschiedene Möglichkeiten der Weiterentwicklung der vorgestellten Methoden und Vorgehensweisen. Dazu werden hier kurz drei angerissen, die für eine reale Implementierung benötigt werden. Hierzu zählt die Anwendung, welche als Thick oder Thin Client ausgebaut werden kann, Verschlüsselung von Daten und Verbindungen und Informationsfusion von verschiedenen Skalen und Gradermittlungsmethoden der Fazialisparese, damit ein vernünftiges und wertungsfreies Ergebnis erzielt werden kann.

Thick- und Thin-Client Architektur Bei einem Thin Client handelt es sich um einen Computer mit reduzierter Leistung, der auf Ressourcen von einem zentralen Server, innerhalb oder außerhalb des Netzwerkes, zugreift. So können die Kostenanschaffungen für den Speicher und leistungsfähigere Prozessoren reduziert werden, da der Server die anspruchsvollen Berechnungen vollzieht (siehe Tabelle 16). Thin Clients verbinden sich über Remote Access oder senden einen Request an den Server für die Ressource, die sie benötigen. Dieser Lösungsansatz bietet auch die Möglichkeit für Andorid und iPhone Betriebssysteme auf Smartphones, die Ermittlung des House-Brackmann Grades auszuführen. Apps auf Smartphones haben nicht die Kapazitäten, Bildverarbeitung mit Neuronalen Netzen zu verarbeiten. Diese würden dem Prozessor für längere Zeit beanspruchen und eventuell zu Überhitzung und starkem Akkuverbrauch führen. Der zentrale Server führt für jeden Request die Detektion aus. Nach dem Prozess wird die Lösung an den Anforderer zurückgesendet. Loadbalancing von den eingesetzten Prozessorkernen und Grafikkarten ist ebenso ein wichtiges Thema. Je nachdem, wie viele Anfragen an einen Client oder auch eine Liste an Patienten die Detektion des Grades der Fazialisparese ausgeführt werden soll, ist es sinnvoll, die Last gleichmäßig zu verteilen. Mithilfe eines Schedulers muss die Last auf die vorhandenen Prozessoren und Grafikkarten verteilen werden. Der Scheduler verwaltet und kennt die maximale Auslastung von den Prozessoren und den Grafikkarten. Im Optimalfall sollte so, wenn eine Häufung von Anfragen bearbeitet werden soll, gleichmäßig auf alle Kerne und GPU's, verteilt werden. So kann das warten die Auftragssteller auf die Rückantwort, das den Grad nach House-Brackmann zurückliefert, reduziert werden.

Vorteile	**Nachteile**
Aufs Nötigste reduziert	Nur mit Netzwerkverbindung nutzbar
Weniger störanfällig	Abhängigkeitsverhältnis vom Server
Kostengünstig	
Niedriger Administrationsaufwand	
Wartungsarm	
Einfach nutzbar	
Hohe Verfügbarkeit	

Tabelle 16: Vor- und Nachteile einer Thin Client Architektur.

Vorteile	Nachteile
Offline Funktionalität	Wartungsintensiv
Direkte Verarbeitung der Eingaben	Hoher Administrationsaufwand
	Kostenintensiv
	Verwundbarkeit durch Ausfall
	Langsam durch Kapazitätsbegrenzungen

Tabelle 17: Vor- und Nachteile einer Thick Client Architektur.

Thick Client oder auch als Fat Client bekannt, sind vollumfänglich ausgestattete, leistungsfähige Computer, die mit ausreichender Rechenkapazität und Speicher, Berechnungen direkt ausführen können. Diese Computer verfügen auch über eine Benutzerschnittstellen, worüber der Anwender die ausgewählte Applikation verwenden kann. Fat Clients werden über Desktop-Computer umgesetzt. Nachteilig dabei ist, dass ein hoher administrativer Aufwand besteht. Alle Clients, die eine neue Softwareversion benötigen, müssen einzeln das Update vornehmen. Einen ebenso wichtigen Aspekt bilden die Kosten. Für eine effiziente Berechnung der Detektion von dem House-Brackmann Score werden Grafikkarten benötigt, die teuer sind. Daher ist der Kosten-Nutzen-Faktor viel zu hoch, um allein auf eine Thick Client basierte Applikationsverwaltung zu setzen (siehe Tabelle 17).

Denkbar wäre eine Kombination aus Thick und Thin Client. Wenn der Server, worauf die Anbindung zur Applikation läuft, ausfällt, kann auf eine Offlineversion der selbigen zugegriffen werden (siehe Tabelle 17). So ist eine ständige Detektion möglich. Nachteilig ist dabei, dass die Berechnungen für die Applikation, je nach Ausstattung des Desktop-Computers, langsamer verläuft. So kann je nach Anforderung sichergestellt werden, dass immer eine Ausführung der Detektierung erfolgt, ausgenommen vom zeitlichen Faktor. Thick und Thin Clients sind beide durch eine API umsetzbar, welche die Sourcen zur Berechnung des Grades beinhaltet. Diese wird dann entweder lokal oder serverseitig gehostet. Per Browser wird die API sodann zugreifbar für den Anwender gemacht.

Verschlüsselung und HTTPS Verbindung Um der Problematik zu entgehen, unverschlüsselt Patientendaten über ein öffentliches Netz zu versenden, sollte darüber nachgedacht werden, vor der API Anwendungssoftware, kurz Applikation (APP), einen Proxy zu implementieren, der den Ein- und Ausgangsverkehr nach den Empfehlungen des Bundesamtes für Sicherheit in der Informationstechnik (BSI) verschlüsselt. So wird sichergestellt, dass Datenpiraterie kein Raum gegeben wird und die vertraulichen Bildmaterialien nur befugten Parteien zugänglich gemacht werden [17][18].

Für eine sichere Verbindung (HTTPS) wird ein Transport Layer Security (TLS) Handshake zwischen dem Client und den Server durchgeführt (siehe Abb. 28). Dazu wird mithilfe des Diffie-Hellman-Schlüsselaustausch Protokolls auf sicherem Wege ein Public-Key-Kryptoverfahren angewendet. Für einen bestmöglichen Schutz sollte die Version TLS 1.2 oder 1.3 verwendet werden, sodass sichere Hashalgorithmen zu Anwendung kommen. Dies wird empfohlen, da bei Protokollversion 1.1 keine kollisionsresistente Hashfunktion (SHA-1) angewendet wird. Auch muss in Betracht gezogen werden, die

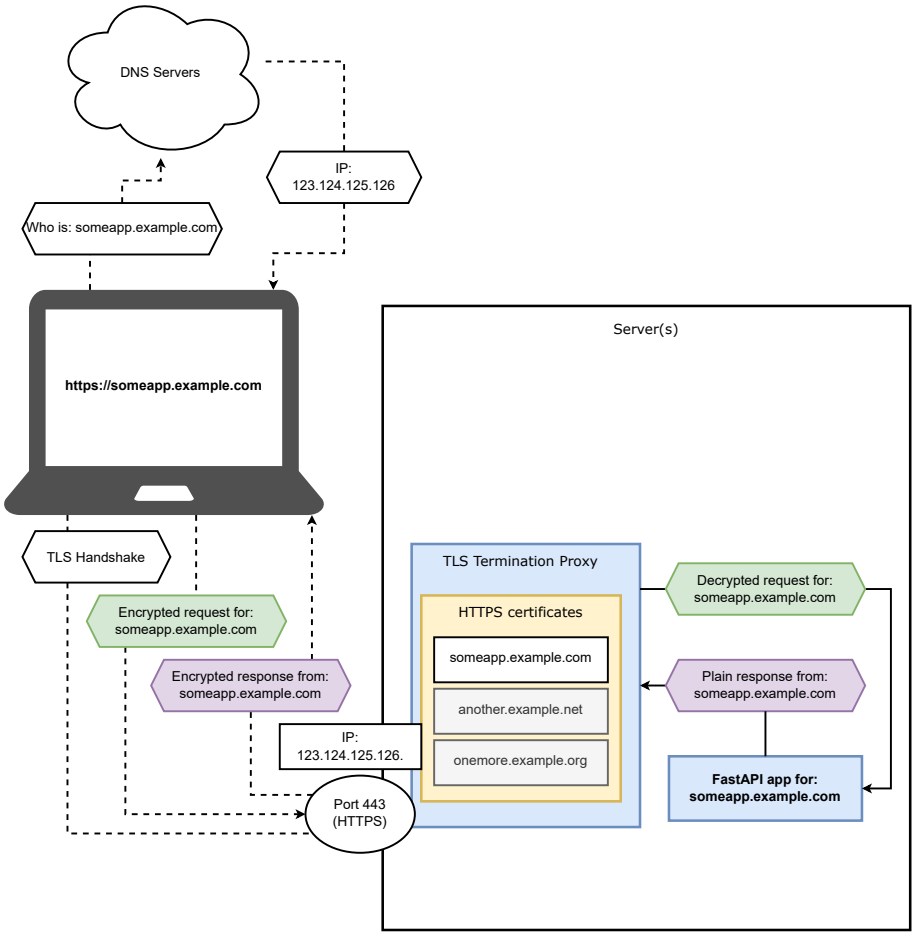


Abbildung 28: Verlauf eines HTTPS Requests im Zusammenhang mit der Applikation (APP). Über ein TLS Handshake werden zuerst Schlüssel zur Verschlüsselung ausgetauscht. Durch die können die restlichen Anfragen sicher zwischen Server und Client transferiert werden. Als Host für die API wurde das Framework von FastAPI betrachtet [17].

Patientendaten nach dem Versenden lokal auf dem Server zu verschlüsseln, sodass ein Fremdzugriff serverseitig ausgeschlossen ist. Damit der Schlüssel außerhalb des Systems nicht einsehbar ist und nur mit einem sehr großen Aufwand berechnet werden kann, jedoch vom System für die Decodierung des Bildmateriales verfügbar ist, sollte dieser während der Laufzeit berechnet werden. Dafür eignet sich ein symmetrisches Kryptoverfahren wie Advanced Encryption Standard (AES) mit einer Schlüssellänge von 256 Bits und von einem Zufallsgenerator erzeugten Schlüssel, der sich für jede*m Nutzer*in individuell generiert und, soweit möglich, nur einmal angewendet wird [18].

Nach der Beendigung und Rücksendung der Ergebnisse der Grade sollten sodann die Patientendaten serverseitig gelöscht werden.

Informationsfusion von verschiedenen Skalen Vorstellbar ist auch, eine anderweitige Ermittlung zusätzlich zu der hier beschriebenen Vorgehensweise zu implementieren. Dazu kann die Sunnybrook Skala zur Anwendung kommen. Diese führt die Detektion separat, anhand derselben neun Bilder aus. Die Sunnybrook Skala verfügt über ein punktebasiertes System, anhand dessen die Grade ermittelt werden. Es kann so ein detaillierteres Ergebnis abgedeckt werden, da die Unterteilung präziser ist als bei der House-Brackmann Skala. Durch eine eventuelle eingeführte Gewichtung der einzelnen Skalen, kann so im nächsten Schritt ein Mittel zwischen der Sunnybrook und House-Brackmann Skala gezogen werden, welches näher am wahren Grad des/der Patient*in liegt. Das Prinzip ist auch bekannt als Informationsfusion oder in der Robotik als Sensorfusion. Hierzu werden verschiedene Sensoren und Daten überlagert. Der Vorteil dabei ist, wenn beide, Sunnybrook und House-Brackmann Skala, schlechte Ergebnisse liefern, gemeinsam dennoch eine präzisere Angabe des wahren Grades der/des Patient*in machen können, als getrennt. Je mehr Skalen und verschiedene Implementierungsmöglichkeiten zur Detektion der Einteilung in die Grade fusioniert werden, desto aussagekräftiger und präziser kann der Grades festgestellt werden.

Die vorgestellten Ideen sollen zur weiteren Anregung in diesem Fachbereich dienen, Zeit zu investieren, um ein System zu erstellen, welches eine unabhängige Bewertung der Fazialisparese im Praxisalltag erlaubt. Das würde ermöglichen, Sprechstundenzeiten von Ärzt*innen zu reduzieren und eine qualitativ hochwertige und unabhängige Meinung des Schweregrades nach House-Brackmann bereitstellen, was den/die Patient*innen zu einer guten Einschätzung ihrer/ihrem Fazialisparese helfen kann. Dazu wird eine sichere und echtzeitkritische Implementierung für die Praxisanwendung benötigt, welche die DatenschutzGrundverordnung (DGSVO) und Rechte der/die Patient*innen berücksichtigt.

Literatur

[1] I. Schmid. „Ff“. In: *Ambulanzmanual Pädiatrie von A-Z*. Springer Berlin Heidelberg, S. 177–207. URL: https://doi.org/10.1007/978-3-662-58432-3_6.

[2] B. für Gesundheit. G51.0: Fazialisparese. URL: https://gesund.bund.de/icd-code-suche/g51-0 (besucht am 20. 01. 2022).

[3] O. Mothes, L. Modersohn, G. F. Volk, C. Klingner, O. W. Witte, P. Schlattmann, J. Denzler und O. Guntinas-Lichius. Automated objective and marker-free facial grading using photographs of patients with facial palsy. *European Archives of Oto-Rhino-Laryngology* 276(12), 3335–3343, Dez. 2019.

[4] J. W. House und D. E. Brackmann. Facial Nerve Grading System. *Otolaryngology–Head and Neck Surgery* 93(2). PMID: 3921901, 146–147, 1985.

[5] A. Welsch, V. Eitle und P. Buxmann. Maschinelles Lernen. *HMD Praxis der Wirtschaftsinformatik* 55(2), 366–382, Apr. 2018.

[6] M. Kang und N. J. Jameson. „Machine Learning: Fundamentals“. In: *Prognostics and Health Management of Electronics*. John Wiley und Sons, Ltd. Kap. 4, S. 85–109. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119515326.ch4. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119515326.ch4.

[7] J. Kleesiek, J. M. Murray, C. Strack, G. Kaissis und R. Braren. Wie funktioniert maschinelles Lernen? *Der Radiologe* 60(1), 24–31, Jan. 2020.

[8] W. Maurer. *Vorlesungsskript Theoretische Informatik*. deutsch. Ostbayerische Technische Hochschule Regensburg, Fakultät Informatik und Mathematik, Inhalt der Vorlesung der Theoretischen Informatik, 1. Semester. 27. März 2016.

[9] C. Beleites, R. Salzer und V. Sergo. Validation of soft classification models using partial class memberships: An extended concept of sensitivity & co. applied to grading of astrocytoma tissues. *Chemometrics and Intelligent Laboratory Systems* 122, 12–22, 2013.

[10] T. Wang, S. Zhang, L. Liu, G. Wu und J. Dong. Automatic Facial Paralysis Evaluation Augmented by a Cascaded Encoder Network Structure. *IEEE Access* 7, 135621–135631, 2019.

[11] J. J. Soraghan, B. O'Reilly, S. He und S. McGrenary. Automatic facial analysis for objective assessment of facial paralysis. In: *1st International Conferfence on Computer Science from Algorithms to Applications (CSAA-2009)*, Dez. 2009.

[12] H. S. Kim, S. Y. Kim, Y. H. Kim und K. S. Park. A Smartphone-Based Automatic Diagnosis System for Facial Nerve Palsy. *Sensors* 15(10), 26756–26768, 2015.

[13] A. Bulat und G. Tzimiropoulos. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In: *International Conference on Computer Vision*, 2017.

[14] T. Contributors. PyTorch Documentation. URL: `https://pytorch.org/docs/stable/` (besucht am 23. 02. 2022).

[15] I. Song, N. Y. Yen, J. Vong, J. Diederich und P. Yellowlees. Profiling bell's palsy based on House-Brackmann score. In: *2013 IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE)*, 1–6, Apr. 2013.

[16] M. Sajid, T. Shafique, M. J. A. Baig, I. Riaz, S. Amin und S. Manzoor. Automatic Grading of Palsy Using Asymmetrical Facial Features: A Study Complemented by New Solutions. *Symmetry* 10(7), 2018.

[17] tiangolo (Sebastián Ramírez). Dokumentation von FastAPI. URL: `https://fastapi.tiangolo.com` (besucht am 03. 01. 2022).

[18] BSI. Bundesamt für Sicherheit in der Informationstechnik. URL: `https://www.bsi.bund.de` (besucht am 03. 01. 2022).

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AES Advanced Encryption Standard

APP Applikation

BSI Bundesamt für Sicherheit in der Informationstechnik

DEA Deterministischer Endlicher Automat

DGSVO DatenschutzGrundverordnung

H-B Grad House-Brackmann Grad

LRU-Cache Least Recently Used Cache

NEA Nicht-Deterministischer endlicher Automat

TLS Transport Layer Security

WCET maximale Ausführungszeit (eng. Worst Case Execution Time)